Unfalsified Visual Servoing for Simultaneous Object Recognition and Pose Tracking

Ping Jiang, Yongqiang Cheng[1], Xiaonian Wang, Zuren Feng

Abstract— In a complex environment, simultaneous object recognition and tracking has been one of the challenging topics in computer vision and robotics. Current approaches are usually fragile due to spurious feature matching and local convergence for pose determination. Once a failure happens, these approaches lack a mechanism to recover automatically. In this paper, data-driven unfalsified control is proposed for solving this problem in visual servoing. It recognizes a target through matching image features with a 3D model and then tracks them through dynamic visual servoing. The features can be falsified or unfalsified by a supervisory mechanism according to their tracking performance. Supervisory visual servoing is repeated until a consensus between the model and the selected features is reached, so that model recognition and object tracking are accomplished. Experiments show the effectiveness and robustness of the proposed algorithm to deal with matching and tracking failures caused by various disturbances, such as fast motion, occlusions and illumination variation.

***Index Terms*— Visual servoing, visual tracking, object recognition, supervisory control, unfalsified adaptive control.**

I. Introduction

3D object recognition and pose tracking has been an active research area with a wide range of applications, such as in visual servo control [1, 2] and marker-less augmented reality [3, 4]. Visual servo control refers to the use of computer vision data to control the motion of a robot for manipulation in a 3D world. It involves techniques from image processing, computer vision, and control theory for real-time robotic operation. There are two different approaches depending on different feedback information used in the control loop [5]. One is position-based visual servo control (PBVS), in which a set of 3D parameters must be estimated from image measurements and controlled to a desired 3D pose. The other is image-based visual servo control (IBVS), in which a set of image features are extracted from image and controlled to the desired features in a reference image. As a result of better real-time performance, IBVS has been introduced into augmented reality for simultaneous tracking and pose detection of a 3D object, known as virtual visual servoing [3]. Instead of physical motion, a feedback controller is designed to drive a virtual model or camera to reduce the image difference between the projections from the 3D model and those sampled in images. Similar to the 3D pose identification in PBVS, it is a dynamic algorithm which tracks and positions a 3D object through aligning and matching, but it relies on real-time feedback control. When the image error is eliminated, the virtual model and the real object get overlapped in three-dimensional space, namely pose estimation of the real object is achieved.

Although model based 3D object recognition and pose registration technologies have been developed with success in many practical applications, such as marker-less augmented reality [6], visual servoing on unknown objects [7], and monocular SLAM (Simultaneous Localization and Mapping) [8], they face robustness issues in less controlled environments, such as the local convergence of matching algorithms. To minimize feature errors between a projected model and a captured image, Gauss-Newton or Levenberg-Marquardt algorithms are often used [9], which can only converge locally and rely on a good initial pose guess. In visual servo control, the developed control algorithms can only guarantee local asymptotic or local convergence due to the nonlinearity of perspective projection [5, 10], which means the system could be trapped in a local minimum but not converge to the desired 3D pose. In practice, it has been observed that the convergence cannot be achieved if the camera displacement has a large orientation error, such as 30 degrees on each axis as reported in [3]. Therefore most existing algorithms require a manual initialization to ensure the initial pose is in the convergence region. The problem of local convergence makes the current algorithms very fragile to disturbance, examples being spurious feature correspondence, fast moving speed, or occlusions, which can result in the failure of model matching and has to be manually corrected by pose re-localization. Keyframes are often used to provide local references for global localization [11], which requires massive memory for storage and expensive computation for keyframe matching. In this step, it is very likely that mismatching of features could happen. Some robust pose estimation algorithms can be used to remove spurious matching, such as consensus based RANSAC [12], which may involve intensive computation for determining consensus poses of a set of randomly selected features. The

[1] Corresponding author: phone +44 1482 466572 Fax +44 1482 466666.

Ping Jiang and Yongqiang Cheng are with Department of Computer Science at University of Hull, HU6 7RX, UK (e-mail: p.jiang@hull.ac.uk, y.cheng@hull.ac.uk, b.peach@2010.hull.ac.uk).

Xiaonian Wang is with Department of Information and Control at Tongji University, China 200092 (e-mail: dawnyear@tongji.edu.cn).

Zuren Feng is with the State Key Laboratory for Manufacturing Systems Engineering, Systems Engineering Institute, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: fzr9910@mail.xjtu.edu.cn)

slow initialization can cause further difficulty on feature correspondence of a dynamic scene in the next sampled image due to the so called wide-baseline matching problem [13], so that the re-initialization of tracking fails again. A challenge faced today for 3D recognition and tracking is to develop technology that can track a moving object in real-time and recover automatically if the tracking fails for some reason [9], for example if the motion is too fast, occlusion occurs, or simply if the target object moves momentarily out of the field-of-view.

In this paper, unfalsified visual servoing is proposed to provide a mechanism for adaptive tracking and online recovery from failures. First, a simple IBVS is developed, which is able to track image features in real-time with local stability. Considering the noise and uncertainties involved in feature extraction, such as those due to background clutter, occlusions, or illumination variation during the tracking, the extracted features from a sampled image may fail to match with their associated features on a model. Therefore a feature detector can only determine a set of candidates in the image possibly belonging to the model. Inspired by the data-driven unfalsified control [14], a supervisor is then introduced on top of the visual servo to select features from the candidate set based on their tracking performance. If the tracking history of some features violates the local convergence of the controller, those features will be falsified and more reliable features will be selected for tracking. If there are too few unfalsified features for reliable tracking, a global search will be automatically started for object recognition and pose estimation in the image. The supervisor can then switch from IBVS to PBVS for reinitializing tracking from the new pose. Therefore the supervisor is a mechanism to make the visual servoing controller aware of any failure and enable its recovery.

The rest of this paper is organized as follows. Related work is reviewed in Section II. A locally stable visual servoing controller is presented in Section III, including a proof of its stability and an induced detectable performance index in order to evaluate the quality of feature matches. Section IV presents an unfalsified adaptive mechanism as the supervisor to the visual servoing controller. It can automatically select the optimal features for model matching and pose tracking according to the performance index. Two experiments are presented in Section V. The first is a simple but comprehensive example to show the main idea and contributions in comparison with other reported methods. The second experiment is a real tracking example to demonstrate its simultaneous recognition and tracking capabilities in a cluttered environment with occlusions and illumination change. Finally, Section VI draws a conclusion.

II. Related Work

In the past few years, there was a particular research interest in robust visual servoing for extending its application area from better structured and controlled industrial environments to real-world natural environments. The uncertainties in a visual servoing loop are mainly from camera-robot models and visual perception. In terms of uncertain or even unknown camera-robot models, adaptive visual servoing has been developed to carry out online model identification, which can be classified as model-based or model independent schemes. In the model-based schemes, an analytic model of a camera-robot system, such as a pinhole camera model [15, 16] or a central catadioptric camera model [17], should be available, or at least partially available. An adaptive algorithm is then used to dynamically estimate the model parameters and drives the robot toward a posture exhibiting the desired image features. However, it is often difficult to estimate the depths of feature points, which are required for image Jacobian matrix calculation. Poor estimate of the depths may result in instability of the control [17]. It may also face a singularity problem due to loss of rank of the estimated Jacobian matrix [15], so that the system may become divergent or fall into a local minimum. To control a vision-guided robot with less prior knowledge, model-independent visual-servoing schemes are proposed [18-20], where the camera-robot model is approximated by a linearized affine model or a neural network and the Jacobian matrix is identified online. However, the estimated image Jacobian may deteriorate so that the control may face the singularity problem. For instance, identification based on Broyden's method [18, 20] can only converge when the initial Jacobian estimation is close to the true matrix. In order to avoid the singularity problem, weight correction for general neural network approximation [19] and Nussbaum gain for control gain exploration [21] were proposed for the stabilization of visual servoing without any camera-robot model knowledge. In addition to the identification based approach, the unmodelled dynamics due to uncertain camera-robot models can also be estimated by a state observer that is then used to compensate the control to reduce the impact of the uncertainties [22]. In the above visual servoing methods, uncertainties are estimated and controlled by observable image errors, which makes an assumption that there is no difficulty obtaining the image errors by matching desired or predicted features with features detected in the image.

Apart from camera-robot models, visual servoing faces ambiguity in visual perception and feature correspondence, especially in a natural scene without artificial markers. Because such ambiguity is in the sensing component of a visual servoing system, the consequence of the caused failures could be very severe, such as large control error, jitter during the tracking, or sudden divergence. Practical visual servoing tends to face such failures due to the complexity of a 3D scene and inherent nonlinear characteristics of a camera, e.g. a practical pin-hole camera has only a limited field-of-view so that the feature points' observation may be lost during the control. Usually IBVS is more suitable for controlling feature points not leaving the field-of-view, but it faces more control difficulties than PBVS such as control singularity and local minimum. A switched-system control between IBVS and PBVS was proposed in [23], which can achieve stability in both pose space and image space simultaneously so that no feature ever leaves the image. A robust PBVS that can achieve global stability and satisfy the field-of-view constraint, even with uncertain camera parameters, was proposed in [24]. In

practical applications, it may not need to keep all features in the field-of-view and some features could appear in or disappear from the image for optimal trajectory tracking. An approach to use weighted features was proposed in [25] to consider dynamic visibility of features during visual servoing. Multi-cameras were also introduced to improve robustness to feature loss during the tracking [26], where features extracted from different cameras were weighted in a Kalman-based sensor fusion approach for visual tracking. Visual servoing research was originated from industrial applications, such as for manipulator positioning in a production line. Providing fiducial markers to reduce sensing uncertainties seems to be more acceptable. For example, the above papers used point fiducials in their experiments, rather than natural features extracted from images. If natural features, such as interest points, are used for visual servo in a real-world environment, it may face more severe robustness issues since spuriously matched features could reach a very high ratio in the overall features and deteriorate the tracking. It is crucial to develop recovery mechanisms when matching failure becomes inevitable, which is often the case for visual tracking in an unknown, large and natural environment such as monocular SLAM for 3D map building [8, 27], PTAM (Parallel Tracking and Mapping) for camera pose estimation [28], and SFM (Structure from Motion) for geometric structure estimation[29].

Tracking failure awareness and recovery are important in monocular SLAM and the others. Any tracking failure may cause the created map to be corrupted completely. During tracking, a motion filter, such as extended Kalman filter [8] or Particle filter [30], can be used for estimating pose of the camera in the next frame. The filter confines the search space according to involved uncertainty. As a result, the search and tracking can be fast and accurate if the uncertainty is low. However, if the uncertainty increases significantly due to, for example, sudden motion or occlusion, the filter based local tracking can fail and a global relocalization has to be utilized for recovery. The recent research mainly focuses on fast feature extraction and matching for real-time recovery. Inspired by randomized trees [31], Williams *et al* [27] developed random lists for fast feature matching when tracking failed. Chekhlov *et al* [32] introduced appearance indexing in the context of Haar wavelet coefficients prior to full matching of descriptors. The appearance index can be a coarse estimate of spatial gradients and enables fast relocalization. In order to achieve fast relocalization in a large space with a mobile device, Straub *et al* [33] used LSH (Locality Sensitivity Hashing) for nearest-neighbor search of low-complexity binary features, such as BRIEF [34] rather than the more distinctive SIFT [35]. However, after such feature based matching, the number of bad associations could be large, due to ambiguity of the feature descriptor. It is common that the outliers have to be further excluded by RANSAC for tracking re-initialization, which is relatively time-consuming. Therefore real-time local tracking and off-line global localization are two linked modes to achieve robust visual tracking in a real environment. A mechanism to effectively manage the mode switch has to be developed. The research on monocular SLAM deals with stationary visual landmarks in the environment. The slower relocalization through RANSAC could be not a big issue because the motion of the camera can slow down during the process. As reported, with some simple switching mechanisms, failures can be detected and the tracking can be re-initialized successfully. For example a global search is activated if the matched feature distance goes over a static threshold [32] or if all attempted observations are unsuccessful [27]. However, this paper focuses on object recognition and tracking using visual servo. The interest points on an object could be dynamic related to the environment. Strict processing time is limited by the servoing period. If the offline global relocalization requires the time span of many frames, the object may have moved out of the convergence region and thus cause the tracking to fail again.

According to the above review, the contribution of this paper can be summarized below:

1) In comparison with the robust visual servoing research such as those dealing with uncertain robot-camera models, this paper tackles uncertainties from the critical sensing component in a vison based feedback system, i.e. for feature point extraction and matching. The consequence of sensing failures could be very severe or even catastrophic if wrongly matched pairs are used to take feedback actions. To the best of our knowledge, this is the first paper focusing on feature matching failures and recovery in visual servoing research.
2) In comparison with the visual tracking research, such as monocular SLAM, PTAM, and SFM, the topic of this paper belongs to the intelligent control paradigm, rather than geometric parameter or camera pose estimation through feature matching. It provides a low-level visual feedback controller with higher-level cognitive capabilities, such as visual attention and recognition, by matching three types of context-information: feature description, rigid dynamic model and perspective distortion model.
3) In addition, the proposed supervisory scheme provides low-level visual servo with feature selection and failure recovery capabilities according to their dynamic tracking performance, rather than offline consensus checks for tracking re-initialization. It is more suitable for moving object tracking and recognition because it can respond at the servo level in real-time.

III. Robust Visual Servoing Control with Detectable Performance Index

Define a tool frame F_t attached to a moving object and a base frame F_c attached to a camera. For robot visual servo, the moving object can be the end-effector of a manipulator. This paper takes virtual visual servoing for augmented reality as an example [3] and thus the moving object is a 3D model $M(t,R)$ that will match and track its counterpart in an image stream, where t and R are translation and rotation between F_t and F_c, respectively. Let $r = \theta k$ be a (3×1) vector containing the axis of rotation k and the angle of rotation θ. Then, $\xi = (t; r)$ is a (6×1)

vector containing global coordinates of an open subset $S \subset \mathbb{R}^3 \times SO(3)$.

Suppose a vector of 3D feature points $P^M = [\cdots P_i^M \cdots]$, where $P_i^M \in \mathbb{R}^3$, $i = 1 \ldots n$, on the rigid model $M(\xi)$ with image projections $p^M = [\cdots p_i^M \cdots]$, $p_i^M \in \mathbb{R}^2$ and corresponding feature descriptor $f^M(p^M) = [\cdots f_i^M \cdots]$. In fact, image features can be any descriptors facilitating robust detection and correspondence in a sampled image, for example SIFT interest points, edges, shapes and textures. With interest points as features, we can obtain the change of the image features caused by the model's motion:

$$\dot{p}^M(\xi) = \frac{\partial p^M}{\partial \xi}\dot{\xi} = J(\xi)u \quad (1)$$

where $J(\xi) \in \mathbb{R}^{2n\times 6}$ is the image Jacobian matrix and $u = \dot{\xi}$ is the velocity of the model M, i.e. the control input of system (1). The objective of this control is to make the feature points in p^M, or part of it, be coincident with detected features p in an image, so that the modeled object M can be recognized from the image and its 3D pose ξ can be determined. We make the following assumptions:

(A1) There are at least 3 features and their corresponding Jacobian matrix $J_{1,2,3}(\xi) \in \mathbb{R}^{6\times 6}$ is not singular;

(A2) $\|J(\xi)\|$ is bounded.

Assumption 1 says, in order to compute the control input, at least 3 non-collinear feature points are required to avoid control singularity [19]. It is known from the PnP problem that there could be four possible poses even if three image feature points are perfectly matched [12]. Therefore, more features have to be used in order to determine correct object pose. As reported in [12], $n\geq 4$ is required for features on a planar surface and $n\geq 6$ is required for general 3D situations. Assumption 2 is for practical applications that require the 3D object to be away from the image plane with depth Z>0.

In a sampled image, p_i^M matched features p_i can be detected by maximizing a similarity measure $Sim(f_i, f_i^M), i = 1 \ldots n$, such as normalized cross-correlation. According to an evaluation score S, three points with the highest values are selected into a triple T:

$$T = [p_1 \quad p_2 \quad p_3] \quad (2)$$

The score needs to consider the similarity of feature matching and spatial distribution of the three points in order to increase control robustness. High similarity measure is preferred because it indicates reliability of feature correspondence. In terms of spatial distribution of the three points, they are expected to be distant from each other to increase the signal-to-noise ratio of the measurement. In addition, three points close to being collinear should be avoided since such an ill-conditioned configuration will lead to a singular Jacobian matrix for control calculation. The tube-collinearity test [36] was used to score the spatial distribution:

$$C(p_1, p_2, p_3) = \prod_{ij}\left(1 - e^{-\frac{1}{2}\left(\frac{d_{ij}}{d_t}\right)^2}\right)$$

where d_{ij} denotes the distance from the i^{th} point, $i \in \{1,2,3\}$, to the j^{th} line formed by the two other points and d_t is the radius of the tube.

The score to select the three points in (2) can be defined as

$$S = Sim(f_1, f_1^M)Sim(f_2, f_2^M)Sim(f_3, f_3^M)C(p_1, p_2, p_3) \quad (3)$$

With three features in (2) that are likely to be correctly matched with the model, a robust method to determine the 3D pose is to use RANSAC [12], which examines the compatibility of other feature points in p with the pose suggested by the triple. If there are enough consensus features in p, the pose is accepted as the initial pose for tracking under model (1); otherwise another triple needs to be selected for the consensus test. The number of total trials can be about 2.0E(k)~ 3.0E(k), where the expected number of trials E(k)=w^{-n} with w to be the probability a detected feature is within the error tolerance of the model and n to be the number of features used to estimate the 3D pose. In our case, n=3 for solving P3P and assume the probability of having a satisfied feature w=0.5; we may need $3w^{-n}$ =24 trials to determine the pose. For tracking of a moving object, slow response due to many trials could cause problems for feature correspondence in the next sampled image, which may change a much easier short-baseline matching to a more difficult wide-baseline matching. In this paper, the consensus trials are embedded into a dynamic visual servoing loop to endow the feedback control with real-time feature selection and tracking capabilities, via the consideration of control performance.

For a triple T in (2), four possible poses ξ_i, i=1..4, can be obtained by solving the P3P problem[12]. Under a pose ξ_i, the visibility set $P^M(\xi_i)$ of the model is first determined, i.e. all feature points in the vector that are visible. The visibility of point $P_i^M(\xi_i)$ is verified by examining if the dot product $P_i^M(\xi_i) \cdot n(i) < 0$, where $n(i)$ is the normal vector of the surface at P_i^M. Then a *candidate set* $p(\xi_i)$ can be determined by selecting the matched features that are within an error tolerance of $p^M(\xi_i)$, i.e. $\| p^M(\xi_i) - p(\xi_i)\| \leq \epsilon_1$, where the tolerance ϵ_1 has taken into account the digital errors of the P3P solution and tracking errors due to processing delay. The pose with most members in it will be used to initialize the tracking, denoted as ξ_0.

Define a task function for visual servoing $s(\xi) = J^T(\xi)e(\xi)$ with tracking error $e(\xi) = p(\xi^*) - p^M(\xi)$, where ξ^* is the unknown object pose. We introduce a modified task function with a deadzone in order to consider robustness of visual servoing to uncertainties[15]:

$$s_\Delta(\xi) = s(\xi) - \varphi(\xi) \quad (4)$$

where $\varphi(k) = \left[\epsilon_0 sat(\frac{s_1(\xi)}{\epsilon_0}) \quad \cdots \quad \epsilon_0 sat(\frac{s_6(\xi)}{\epsilon_0})\right]^T$ with deadzone width ϵ_0 and saturation function $sat(.)$.

Proposition 1*:* For $p^M(\xi)$ in (1) to track the matched $p(\xi^*)$ in a sampled image, visual servoing law $u = \left(J^T(\xi)J(\xi)\right)^{-1}Ks(\xi)$, $K > 0$, can guarantee $s_\Delta(\xi)$ converging exponentially into the deadzone ε_0 with $\int_0^t \|s_\Delta(\xi)\|^2\, dt \leq \gamma_0^2$, where γ_0^2 is a positive constant, if $K \geq \left\|J^T(\xi)e(\xi) + J^T\dot{p}(\xi^*)\right\|_\infty / \varepsilon_0$.

Proof: Define a Lyapunov function $V(t) = 1/2 s_\Delta^T(\xi)s_\Delta(\xi)$. From definition (4),

$V(t) = 1/2\sum_{j=1}^{n} s_{\Delta j}^2(\xi)$, where $s_{\Delta j}^2(\xi) = \begin{cases} (s_j - \varepsilon_0)^2 & s_j > \varepsilon_0 \\ 0 & |s_j| \le \varepsilon_0 \\ (s_j + \varepsilon_0)^2 & s_j < -\varepsilon_0 \end{cases}$

and $s_{\Delta j}^2(\xi)$ is differentiable.

Then we have $\frac{\mathrm{d}}{\mathrm{dt}}\left(s_{\Delta j}^2(\xi)\right) = 2s_{\Delta j}(\xi)\dot{s}_j(\xi)$, i.e. $\dot{V}(t) = s_\Delta^T(\xi)\dot{s}(\xi)$.

$$\dot{V}(t) = s_\Delta^T(\xi)\dot{s}(\xi)$$
$$= s_\Delta^T(\xi)\left(\dot{J}(\xi)^T e(\xi) + J^T(\xi)\dot{p}(\xi^*) - J^T(\xi)\dot{p}^M(\xi)\right)$$

From (1), we have

$$\dot{V}(t) = s_\Delta^T(\xi)\left(\dot{J}^T(\xi)e(\xi) + J^T(\xi)\dot{p}(\xi^*) - J^T(\xi)J(\xi)u\right)$$

Let the control

$$u(p^M(\xi), p(\xi^*)) = \left(J^T(\xi)J(\xi)\right)^{-1}Ks(\xi) \quad (5)$$

where K>0 and $J^T(\xi)J(\xi)$ is nonsingular according to (3) and (A1). From (4) we have

$$\dot{V}(t) = s_\Delta^T(\xi)\left(\dot{J}^T(\xi)e(\xi) + J^T\dot{p}(\xi^*) - K(s_\Delta(\xi) + \varphi(\xi))\right)$$
$$= s_\Delta^T(\xi)\left(\dot{J}^T(\xi)e(\xi) + J^T\dot{p}(\xi^*) - Ks_\Delta(\xi) - \epsilon_0 K\mathrm{sgn}(s_\Delta(\xi))\right)$$
$$\le \|s_\Delta(\xi)\|_1\left(\left\|\dot{J}^T(\xi)e(\xi) + J^T\dot{p}(\xi^*)\right\|_\infty - \epsilon_0 K\right) - K\left\|s_\Delta(\xi)\right\|^2$$

where $\mathrm{sgn}(s_\Delta(\xi)) = [\mathrm{sgn}(s_{\Delta 1}) \ \cdots \ \mathrm{sgn}(s_{\Delta n})]^T$ is the sign vector and $\|\cdot\|_1, \|\cdot\|$ and $\|\cdot\|_\infty$ are L^1, L^2 and L^∞ norms respectively.

If $K\varepsilon_0 \ge \left\|\dot{J}^T(\xi)e(\xi) + J^T\dot{f}(\xi^*)\right\|_\infty$, $\dot{V}(t) \le -K\left\|s_\Delta(\xi)\right\|^2 = -2KV(t)$ and $s_\Delta(\xi)$ converges exponentially into the deadzone ε_0 with $\int_0^t \|s_\Delta(\xi)\|^2\,dt \le \gamma_0^2$, where γ_0^2 is a positive constant. □

Equation (5) is a conventional visual servoing control that can converge the task function into the deadzone with robustness to uncertainties in motion and the projection model. As a result, the local stability of all features, i.e. $\lim_{t\to\infty}\|p^M(\xi,t) - p(\xi^*,t)\| \le \epsilon_3$ if ξ is in the neighborhood of ξ^*, can be guaranteed, similar to the proof in [10]. However, it requires all features to have been matched correctly, i.e. $p_i^M \leftrightarrow p_i$, $i = 1 \ldots n$, where "↔ " indicates a match between a feature point in a model and one extracted from the image. In real applications, it is often the case that spurious features could be matched and used in the control loop, which could significantly deteriorate the tracking performance. Therefore, a detectable performance index needs to be used to evaluate each matched pair for the purpose of feature selection.

For an unknown object pose $\xi^*(t)$, $e(\xi^*, \xi) = p(\xi^*) - p^M(\xi)$ can be linearized by using Taylor series around ξ^*:

$$e = J(\xi)(\xi^* - \xi) + \mathrm{o}(\xi^* - \xi)^2 \quad (6)$$

and $$s = J^T(\xi)J(\xi)(\xi^* - \xi) + J^T(\xi)\mathrm{o}(\xi^* - \xi)^2$$

Therefore,

$$\|s\| \ge \|J^T(\xi)J(\xi)(\xi^* - \xi)\| - \|J(\xi)\|\mathrm{o}\|\xi^* - \xi\|^2$$
$$\ge \lambda\left(J^T(\xi)J(\xi)\right)_{min}\|(\xi^* - \xi)\| - \|J(\xi)\|\mathrm{o}\|\xi^* - \xi\|^2$$

where $\lambda\left(J^T(\xi)J(\xi)\right)_{min} > 0$ is the minimum eigenvalue of $J^T(\xi)J(\xi)$. Hence, the pose difference can be

$$\|(\xi^* - \xi)\| \le \lambda\left(J^T(\xi)J(\xi)\right)_{min}^{-1}(\|s\| + \|J(\xi)\|\mathrm{o}\|\xi^* - \xi\|^2)$$

If the model pose ξ can be initialized into a neighborhood of ξ^* with

$$\mathrm{o}\|\xi^* - \xi\|^2 \le \alpha\lambda\left(J^T(\xi)J(\xi)\right)_{min}\|J(\xi)\|^{-1}\|(\xi^* - \xi)\|, \ \alpha < 1, \quad (7)$$

then

$$\|(\xi^* - \xi)\| \le \beta\lambda\left(J^T(\xi)J(\xi)\right)_{min}^{-1}\|s\|, \text{ where } \beta = \frac{1}{1-\alpha} \quad (8)$$

For the i^{th} matched features of an object, substitute (7) and (8) into (6)

$$\|e_i\| \le \|J_i(\xi)\|\|(\xi^* - \xi)\| + \mathrm{o}\|\xi^* - \xi\|^2$$
$$\le \left(\|J_i(\xi)\| + \alpha\lambda\left(J^T(\xi)J(\xi)\right)_{min}\|J(\xi)\|^{-1}\right)\|(\xi^* - \xi)\|$$
$$\le \beta\left(\|J_i(\xi)\|\lambda\left(J^T(\xi)J(\xi)\right)_{min}^{-1} + \alpha\|J(\xi)\|^{-1}\right)\|s\| \quad (9)$$

From (4), we have

$$\|e_i\| \le \beta\left(\|J_i(\xi)\|\lambda\left(J^T(\xi)J(\xi)\right)_{min}^{-1} + \alpha\|J(\xi)\|^{-1}\right)(\|s_\Delta\| + \|\varphi\|)$$

Because $\|\varphi\| \le \sqrt{6}\|\varphi\|_\infty = \sqrt{6}\varepsilon_0$,

$$\|e_i\| - \beta\left(\|J_i(\xi)\|\lambda\left(J^T(\xi)J(\xi)\right)_{min}^{-1} + \alpha\|J(\xi)\|^{-1}\right)\sqrt{6}\epsilon_0 \le \beta\left(\|J_i(\xi)\|\lambda\left(J^T(\xi)J(\xi)\right)_{min}^{-1} + \alpha\|J(\xi)\|^{-1}\right)\|s_\Delta\| \quad (10)$$

Design a deadzone for $e_{\Delta i}(\xi) = e_i(\xi) - \phi(\xi)$, where $\phi(\xi) = \left[\epsilon_2 sat(\frac{e_{ix}(\xi)}{\epsilon_2}) \quad \epsilon_2 sat(\frac{e_{iy}(\xi)}{\epsilon_2})\right]^T$ with deadzone width ϵ_2 and $e_i = \left[e_{ix}, e_{iy}\right]^T \in \mathbb{R}^2$.

From $\|x\|_\infty \le \|x\| \le \sqrt{N}\|x\|_\infty$ for $\forall\, x \in \mathbb{R}^N$, we have

$$\|e_{\Delta i}(\xi)\| \le \sqrt{2}\|e_i(\xi) - \phi(\xi)\|_\infty$$
$$= \begin{cases} \sqrt{2}(\|e_i(\xi)\|_\infty - \epsilon_2) & if\ \|e_i(\xi)\|_\infty \ge \epsilon_2 \\ 0 & otherwise \end{cases}$$
$$\le \begin{cases} \sqrt{2}(\|e_i(\xi)\| - \epsilon_2) & if\ \|e_i(\xi)\|_\infty \ge \epsilon_2 \\ 0 & otherwise \end{cases}$$

If $\epsilon_2 \ge \beta\left(\|J_i(\xi)\|\lambda\left(J^T(\xi)J(\xi)\right)_{min}^{-1} + \alpha\|J(\xi)\|^{-1}\right)\sqrt{6}\epsilon_0$, from (10), we have

$$\|e_{\Delta i}(\xi)\| \le \sqrt{2}\beta\left(\|J_i(\xi)\|\lambda\left(J^T(\xi)J(\xi)\right)_{min}^{-1} + \alpha\|J(\xi)\|^{-1}\right)\|s_\Delta\| \quad (11)$$

The exponential convergence of s_Δ implies exponential convergence of $e_{\Delta i}$, and the following integration is bounded:

$$\int_0^t \|e_{\Delta i}(\xi)\|^2\,dt \le \sqrt{2}\beta\left(\|J_i(\xi)\|\lambda\left(J^T(\xi)J(\xi)\right)_{min}^{-1} + \alpha\|J(\xi)\|^{-1}\right)\int_0^t \|s_\Delta(\xi)\|^2\,dt$$
$$\le \sqrt{2}\beta\left(\|J_i(\xi)\|\lambda\left(J^T(\xi)J(\xi)\right)_{min}^{-1} + \alpha\|J(\xi)\|^{-1}\right)\gamma_0^2 = \gamma_1^2 \quad (12)$$

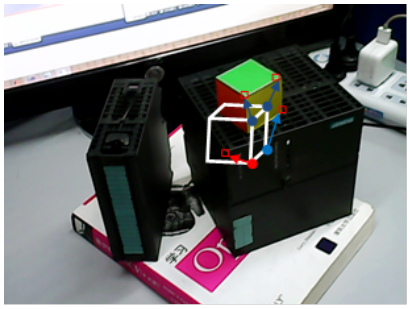
It indicates that a feature point on the 3D model can track its correspondence in the image with a bounded square error integral under the control (5) if the match is correct. Therefore we can use (12) as a performance index to reveal if a matching pair exhibits the expected movement pattern constrained by its model (1) and control (5). The matched feature pairs can then be unfalsified or falsified from the control loop accordingly.

IV. Unfalsified Visual Servoing Control for Feature Selection

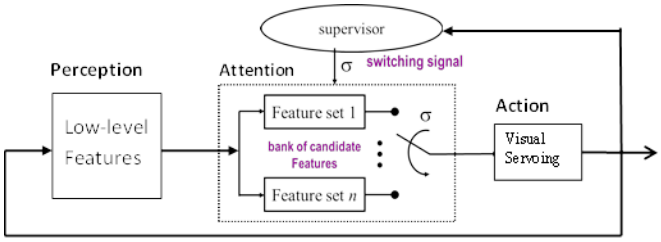
Adaptive control to cope with slow time-varying uncertainties in terms of a known plant structure has been well

developed. When such model structure is less known so that the model uncertainties are large and dynamic, a single adaptive controller cannot be designed to cope with all situations. A high-level supervisor is thus introduced to coordinate a group of candidate controllers with their respective set of possible models for selecting the current best performing controller based on online model validation, i.e. supervisory control [37, 38]. For control of a system without any assumptions on the plant, unfalsified control has been developed in the field of supervisory control [14, 39-41]. An unfalsified controller can be considered as a data-driven supervisory mechanism that selects a feasible controller from a set of candidates in order to meet an expected control performance. It collects input-output data of the unknown plant by inserting different controllers into the control loop to evaluate inconsistencies between the desired control performance and the collected data constrained by the plant. If any inconsistencies are detected, mismatched controllers are gradually falsified from the candidate set, and the remaining unfalsified controllers are feasible to meet the control performance or stability according to the data so far. Therefore, unfalsified control can be designed with less model knowledge and can exhibit superb robustness to control systems with high uncertainties, such as due to large time-variations [42] and system faults [43].

Image feature matching in motion can be very uncertain, due to changed illumination, clutter, or occlusions. The initial features in the candidate set $p(\xi_i)$ could include many spuriously matched features, so that the control in *Proposition* 1 cannot track the 3D object. This paper adopts the concept of unfalsified control for feature selection, which screens the features in the candidate set based on their tracking performance. Instead of controller selection in the conventional sense, it falsifies those spurious features and selects correct and robust features for 3D object tracking in real-time as shown in Fig.1. In Fig.1.a, four corner points on a model, the white cube,



(a) Feature matching with the model, correct in blue and spurious in red.



(b) Feature selection by a supervisor

Fig.1. Unfalsified visual servoing of a 3D model

are matched with extracted feature points in the image with three points matched correctly (blue) but one mismatched (red). It is expected that the mismatched feature can be falsified from the control loop due to its poor tracking performance and the correct corner points can be selected for the tracking. The block-diagram of the proposed unfalsified visual servoing is shown in Fig.1.b. It consists of three components, namely perception, attention and action. The perception block extracts all features from an image. The attention block, based on a supervisor's evaluation, attempts to select a set of features matched with the model for its recognition. The action block then uses a visual servoing controller to drive the model in order to illustrate their tracking behaviors for attracting the supervisor's attention. Therefore, it is an intelligent controller that provides low-level visual servo with context awareness to a scene.

Definition 1: a feature p_i in candidate set $p(\xi^*)$ can be moved into a consensus set $C(\xi)$ if it has converged into the deadzone ϵ_2 from the model feature, i.e. $\| p_i(\xi^*) - p_i^M(\xi) \| \leq \epsilon_2$.

It was shown in (11) and *Proposition* 1 that a correctly matched feature in a candidate set can converge to the deadzone. More features attracted into the consensus set demonstrate higher belief that the modeled object appears in the image and its pose is captured accurately. The features in the consensus set can be further used in control law (5) to drive the model for accurate tracking.

Consider a γ dependent performance specification $T_{spec} = \{p, p^M, u | I(p, p^M, u) \leq \gamma\}$, where γ is a positive bound but unknown. It describes the tracking performance $I(p, p^M, u)$ of a matched feature pair under control u, such as (5). A spuriously matched feature usually causes performance deterioration over time and the cost to exceed the bound γ. Therefore, we can use the performance cost $I(p, p^M, u)$ as a measure to detect spurious features in the candidate set.

Definition 2: A feature $p_i \in p(\xi^*)$ is said to be *falsified* by measurement information if this information is sufficient to deduce that the performance specification $(p_i(t), p_i^M(t), u(t)) \in T_{spec}$ would be violated if the object was controlled by u. Otherwise the feature p_i is said to be *unfalsified*.

From the exponential convergence of $e_{\Delta i}$ in (12), we can evaluate the tracking performance with

$$T_{spec} = \left\{ p, p^M, u \middle| \int_0^t \|e_{\Delta i}(\xi)\|^2 \, dt \leq \gamma_1^2 \right\} \qquad (13)$$

Because this is a local conclusion and true only in candidate region $\|e_{\Delta i}\| \leq \varepsilon_1$, another performance specification can be further defined to see if a feature is admissible. If a feature violates this condition, which is evaluated by the following admissible specification, it can be falsified directly from the candidate set.

$$T_{admi} = \{ p, p^M, u | \|e_{\Delta i}\| \leq \varepsilon_1 \} \qquad (14)$$

Considering the worst case, we assume that at least n features need to be matched in order to ascertain a modeled object found in the image. The following unfalsified visual servoing algorithm is proposed for feature selection by examining their tracking performance and for pose estimation if model matching is ascertained:

Algorithm

1) Search for the triple $T = [p_1 \quad p_2 \quad p_3]^T$ in a sampled image with the highest score and initialize two models with pose ξ_0 by solving the P3P problem, i.e. $M_1(\xi_0)$ and $M_2(\xi_0)$.
   Define consensus radius ε_2 and candidate radius ε_1.
   Determine N visible features of model M_1 (ξ_0) and define the corresponding visible set $V \in \mathbb{N}^N$.
   Let k=1, ξ_1=ξ_0, ξ_2=ξ_0 and $I(i,k-1)$ =0 with $i \in V$. Randomly select n features into unfalsified set $U \in \mathbb{N}^n$ and set consensus set $C = \emptyset$.
2) Sample an image and extract features $p_i(\xi_1)$ with $M_1(\xi_1)$ through *local matching*, where $i \in V$.
3) Drive M_1 with control law $u(T_M, T)$ in (5) to a new ξ_1.
4) Calculate performance indices for every feature $i \in V$:
$$I(i,k) = I(i,k-1) + \int_{(k-1)\Delta t}^{k\Delta t} \|e_{\Delta i}(\xi_1)\|^2 \, dt$$
   where Δt is the sampling interval; $I(i,k)$ is for evaluation of feature i's tracking.
5) Repeat ε-cost minimization [44] n times:
   If $\max_{i \in U} I(i,k) > \min_{j \in V \setminus U} I(j,k) + \varepsilon$
$$U = U \setminus \left\{ \arg\max_{i \in U} I(i,k) \right\} \cup \left\{ \arg\min_{j \in V \setminus U} I(j,k) \right\}$$
   Falsify the features not admissible:
$$U = U \setminus \{ i \,|\, \|e_{\Delta i}\| > \varepsilon_1, i \in U \}$$
   k=k+1.
   Where $\cup$ and $\setminus$ represent set union and complement operation, respectively.
6) Drive M_2 with control law $u(p_{Mi}, p_i | i \in U)$ in (5) to pose ξ_2.
   Determine consensus features: $C = \{ i \,|\, \|e_{\Delta i}\| \le \varepsilon_2, i \in V \}$
7) If $|U \setminus C| + |C| \ge n$
   The model is unfalsified from the image and the visible set V of $M_1(\xi_1)$ is updated.
   If $|C| \ge n$, output the pose ξ_2 as the object pose.
   Go to step 2.
   Otherwise the triple T is falsified and go to step 1 for *global search* of another triple and pose ξ_0.
   Where $|.|$ is the cardinality of a set.

In the algorithm, step 1 is for re-localization when starting initial tracking or after a tracking failure. It can be considered as the recognition mode of the visual servo, which involves finding three best matched features T in an image and initializes tracking. Two models $M_1(\xi_0)$ and $M_2(\xi_0)$ of the object to be tracked are initialized to pose ξ_0, where $M_1(\xi_0)$ is for feature selection and $M_2(\xi_0)$ is for final visual tracking with the selected features. In the conventional unfalsified adaptive control [44, 45], a fictitious reference signal is calculated by considering that the measured data can be reproduced exactly under the control of a candidate controller. As a result, the performance index for verifying the finite gain stability of a specific controller can be obtained without switching it into the real control loop. However, for feature selection, performance specifications defined in (13) and (14) can only be determined by the control errors of individual feature points. A feature cannot be evaluated without it actually in the control loop. Therefore, we take an open-loop approach by introducing $M_1(\xi_1)$ controlled by the three best matched points in triple T for evaluation of other features' tracking consensus but without their involvement in the control loop.

During the real-time visual tracking, an image is sampled and the feature points matching with the visible keypoints on $M_1(\xi_1)$ are first detected through *local search*, as in step 2. It can be implemented through a filter based prediction and matching process, e.g. a Kalman filter based interest point prediction and a local correction in a window around the predicted point. Then $M_1(\xi_1)$ is driven by (5) with visual servoing of the three best matched features in T. The tracking performance and admissibility of all other matched features are evaluated by (13) and (14), respectively, as in step 4 and step 5. Step 5 selects n best tracked points so far with minimum error integration in (13), passing through an ε-hysteresis, into the unfalsified set U, where the ε-hysteresis is introduced to avoid infinite switching. If a selected feature point violates the admissible condition in (14), corresponding to an unreliable matching, it will be eliminated from U. Then all unfalsified features in U will be used to drive M_2, by visual servoing control (5). The number of consensus features, i.e. with $\|e_{\Delta i}\| \le \varepsilon_2$, is counted. If this number is greater than n, tracking has been accurate enough to output the pose of the object. Even if there are not enough consensus features detected in the image but there are sufficient unfalsified features, so that $|U \setminus C| + |C| \ge n$ in step 7, the target may still be in the image and visual tracking continues. Otherwise, the current triple fails and a global relocalization is initialized. A new triple with the next highest score is selected to determine the new pose ξ_0 for the next tracking trial.

Proposition 2: For a given 3D model, assume at least n image features can be matched into a tolerance ε_2 if it appears in an image from any viewpoint. If triple T in the algorithm consists of three correctly matched features and the initial pose ξ_0 can be in a neighborhood of ξ^* satisfying (7), the algorithm can always converge to n correctly matched features with a finite number of feature switches. The object pose can be estimated with accuracy $\|(\xi^* - \xi)\| \le \sqrt{6}\beta\lambda\left(J^T(\xi)J(\xi)\right)^{-1}_{min}\varepsilon_0$.

Proof:

Let M_1 under control $u(T_M, T)$ in (5). If the three features in T are correctly matched, the n correct features on the object should satisfy $I(i^*,t) = \int_0^t \|e_{\Delta i^*}(\xi)\|^2 \, dt \le \gamma_1^2$ from (12). However, any mismatched features have $\lim_{t \to \infty} I(i,t) = \infty$, therefore it is cost-detectable [44]. In addition, $I(i,t)$ is monotone non-decreasing in time. From *Proposition* 1 in [44], the ε-cost minimization will make the algorithm converge to a feature satisfying $I(i,t) \le I(i^*,t) + \varepsilon$ for all t within finitely many feature switches. The overall number of the switches is less than $(N+1)\gamma_1^2/\varepsilon$ with N to be the total number of features. That means $\lim_{t \to \infty} I(i,t) \le \gamma_1^2 + \varepsilon$ has an upper-bound and therefore feature i with $\lim_{t \to \infty} e_{\Delta i}(\xi) = 0$ is a correctly matched feature exhibiting exponential convergence. By repeating

ε-cost minimization n times, n correctly matched features can be obtained with finitely many switches.

If M_2 is controlled by the feedback of the n correctly matched features, from *Proposition* 1, $\lim_{t\to\infty} s_\Delta(\xi) = 0$, i.e. $\lim_{t\to\infty} \|s(\xi)\| \leq \sqrt{6}\lim_{t\to\infty} \|s(\xi)\|_\infty = \sqrt{6}\varepsilon_0$. Substituting it into (8), we have

$$\|(\xi^* - \xi)\| \leq \sqrt{6}\beta\lambda\left(J^T(\xi)J(\xi)\right)^{-1}_{min}\varepsilon_0 \qquad \square$$

In summary, the algorithm proposes a complete process for reliable feature selection, feature tracking, failure detection, and failure recovery, which is coordinated by the supervisor. It first selects the most likely three features as the triple to drive $M_1(\xi_1)$. The evaluation of candidate features on the model $M_1(\xi_1)$ is based on their tracking performances under the visual control (5) with the three features of T in the control loop. If the modeled object appears in the image, *Proposition* 2 states that the ε-cost minimization process can always obtain n correctly matched features from the image with finite number of switching. During this dynamic selection process, an object is unfalsified from the image if the number of unfalsified features plus consensus features is more than n. Further driven by the n unfalsified features, $M_2(\xi_2)$ can eventually converge to a pose with error less than $\sqrt{6}\beta\lambda\left(J^T(\xi)J(\xi)\right)^{-1}_{min}\varepsilon_0$ from the actual pose of the object. Otherwise, a new triple with the next highest score will be used for pose re-initialization and consensus check. From the view of visual servoing, the supervisor switches IBVS of local tracking to PBVS of global positioning if a failure happens.

V. Testing and Experiments

Bad feature-association may often happen when natural features are used in visual servoing. The consequence of such failures could be catastrophic to the feedback control system. The proposed unfalsified visual servoing introduced a supervisor on top of a visual servoing loop to achieve automatic feature selection and failure recovery. Two experiments were used to demonstrate the proposed control scheme. The first one was a toy-example focusing on method presentation and comparison with other visual servoing schemes, where a cube as shown in Fig.2.a was required to be recognised and tracked in a virtual environment. In the second experiment, visual tracking under various adverse conditions was implemented and tested in order to verify the effectiveness of the proposed algorithm for automatic feature selection and failure recovery. A tea box as shown in Fig.2.b was recognized and tracked in a cluttered background with a sampling rate of 20fps. The experiments were developed and implemented by using Halcon HDevelopment environment with a laptop of i7 2.6GHz processor and 8GB RAM.

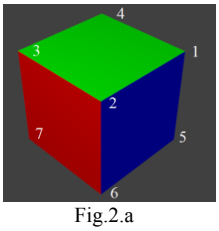


Fig.2.a

Fig.2.b

Fig.2. (a) A cube to be recognized and tracked in section *A*. (b) A tea box to be recognized and tracked in section *B*.

A. Comparative Study

As shown in Fig.3, there are three objects on a table. A robot is asked to find a cube on the table and track its motion in order to pick it up. The visual features extracted from the robot vision are the corner points detected by the Harries detector, as labelled in Fig.3 for example. The feature descriptor of a corner point can be defined as a color vector to describe all face-colors meeting around the point:

CV = [Red Purple Light-blue Yellow Green Blue] (15)

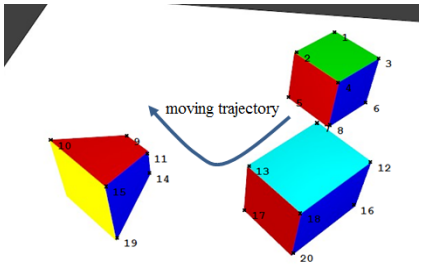


Fig.3. Trajectory of the cube and feature points detected in the image.

The 3D model of a $18mm\times18mm\times18mm$ cube is first created for visual tracking as shown in Fig.2.a. The vertices are candidates for interest point matching.

Therefore, the feature descriptors of eight vertices on the cube model can be given as

$$\begin{aligned} f_1^M &= [0,0,0,1,1,1],\ f_2^M = [1,0,0,0,1,1] \\ f_3^M &= [1,1,0,0,1,0],\ f_4^M = [0,1,0,1,1,0] \\ f_5^M &= [0,0,1,1,0,1],\ f_6^M = [1,0,1,0,0,1] \\ f_7^M &= [1,1,1,0,0,0],\ f_8^M = [0,1,1,1,0,0] \end{aligned} \qquad (16)$$

where "1" indicates a face with the corresponding color meeting around the point under consideration. For example, f_2^M of vertex 2 in Fig.2.a is where a red, a green and a blue face meet.

For recognition and tracking of the cube, an image is sampled and interest points are extracted by the Harris operator, as shown in Fig.3. Around the obtained points, a 10 by 10 window is opened for color clustering. If more than 10 pixels in the window appear a color close to the one in (15), the corresponding element in the feature vector is set. Taking the 2nd point detected in Fig.3 as an example, the red and green faces meet at this point and $f_2 = [1,0,0,0,1,0]$. For visual servoing, the detected feature points need to be matched with those in model (16) through a similarity measure, e.g. normalised cross correlation:

$$Sim(f_i, f_i^M) = \left\langle \frac{f_i}{\|f_i\|}, \frac{f_i^M}{\|f_i^M\|} \right\rangle \qquad (17)$$

A.1: Maximum Similarity Based Match in Visual Servoing

In conventional visual servoing, features on the model and those detected in the image are matched to form image errors if they show the maximum similarity. However such a similarity based matching could be wrong in real applications and fail the tracking, due to 1) ambiguity of feature descriptors and 2) feature occlusions during the tracking. For example, the 6th vertex on the model can be spuriously matched with the 18th in

Fig.3 because $Sim(f_{18}, f_6^M) = 1$ but $Sim(f_8, f_6^M) = 0.8165$, where $f_{18} = [1,0,1,0,0,1]$ and $f_8 = [1,0,0,0,0,1]$. Such a feature ambiguity comes from the invisible light-blue face of point 8.

Conventional visual servo as in (5) was implemented with control gain K=0.5:

$$u(p^M(\xi), p(\xi^*)) = \left(J^T(\xi)J(\xi)\right)^{-1} J^T(\xi)Ke(\xi) \qquad (18)$$

where the image Jacobian matrix involving the n feature points

$$J = [J_1^T \cdots, J_i^T, \cdots J_n^T]^T \in \mathbb{R}^{2n\times 6},$$

for each feature points $p_i = [x_i, y_i]$ with depth Z_i

$$J_i = \begin{bmatrix} \frac{1}{z_i}\begin{bmatrix} f_x & 0 & -x_i \\ 0 & f_y & -y_i \end{bmatrix} & \begin{matrix} -x_i y_i/f_x & f_x + x_i^2/f_x & -y_i \\ -f_y - y_i^2/f_y & x_i y_i/f_y & x_i \end{matrix} \end{bmatrix}$$

with focal length $f_x = f_y = 0.035m$.

Spatial error e can be further used for matching failure detection and recovery through switching between the following two control states:

1) Tracking.

   Under this state, the matches are considered to be correct with root-mean-square error $RMSE(e) \le \varepsilon$. The feedback (18) is applied with error e at frame i. Local tracking is achieved by finding the matched features $p(\xi^*)$ in the next frame i+1 through a local search around the model based prediction p^M. With such a local search, the real-time tracking can be achieved.

2) Searching

   If $RMSE(e) > \varepsilon$, the matching is falsified and a global search should be carried out. The best matched $p(\xi^*)$ with the model can be determined by finding the pairs with the highest similarity measure.

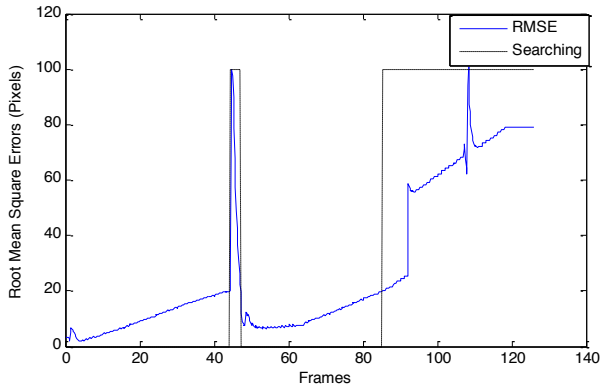


Fig.4.a

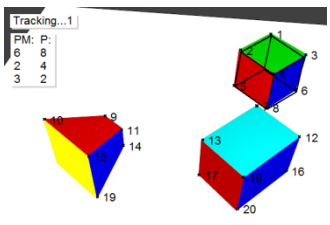


Fig.4.b

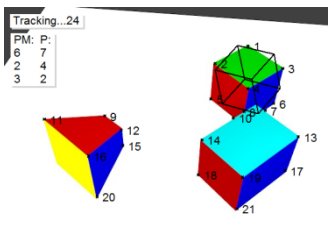


Fig.4.c

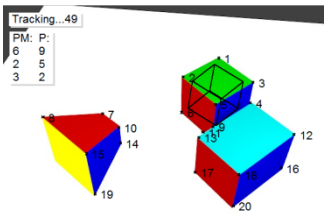


Fig.4.d

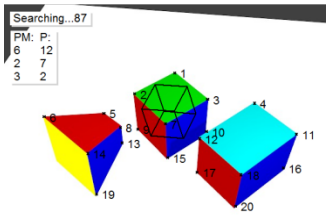


Fig.4.e

Fig.4. Conventional visual servoing, see video Cube_CV.mov. (a) Tracking error. (b) Tracking at frame 1. (c) Tracking at frame 24. (d) Tracking at frame 49. (e) Feature search at frame 87.

Let n=3 and ε =20 pixels. Assume that there is neither pose error nor false matching initially as in Fig.4.b, i.e. $f_6^M \leftrightarrow f_8$, $f_2^M \leftrightarrow f_4, f_3^M \leftrightarrow f_2$. The tracking errors are shown in Fig.4.a, with the dashed line to indicate the control is in the searching or tracking state. The model is illustrated as the black skeleton of the cube and its tracking trajectory can be observed from Fig.4.b to Fig.4.e. Initially the model can track the cube with correctly matched feature points in Fig.4.b. Since the 8th frame, vertex 6 of the model has been occluded by the cuboid. As a result, vertex 6 was wrongly matched with point 7 as in Fig.4.c, which caused gradually increased tracking error. When the RMS error reached 20 at the 45th frame, a global search based on the maximum similarity was carried out. Vertex 6 was matched with feature point 9, which is a spurious match again. After a big transient process, the model converged as in Fig.4.d at frame 49. Because the RMS error is less than ε, it remains in the tracking state and uses the spuriously matched points for visual servo. At frame 87, the RMS control error caused by the spurious matching between vertex 6 and point 12 exceeded the threshold and therefore a best matched feature point pair needed to be searched for again, as shown in Fig.4.e. However, the search was never successful because point 12 keeps the highest similarity to vertex 6 in the model, which is a pair of spurious matching.

A.2: Similarity and RANSAC Based Match in Visual Servoing

From the experiment in section A.1, feature similarity based matching exhibits high uncertainty on feature-association in visual servoing. The consensus based RANSAC [12] was often used to improve the robustness of a relocalisation mechanism for visual tracking, such as in monocular SLAM [27, 32]. As discussed in section III, such a trial-and-error based consensus check could be quite time consuming. The consensus achieved with the image at a specific frame could be violated again if a moving object has changed its pose significantly in this slow consensus check period.

The RANSAC based relocalization is implemented in this section. There are two states in the control flow, i.e. 1) tracking and 2) searching. In the tracking state, conventional visual servoing (18) is applied for local tracking if a consensus has been achieved for a triple, which is examined by checking if the n best matched points can have $RMSE(e) \le \varepsilon$. If the consensus cannot be achieved, i.e. $RMSE(e) > \varepsilon$, the supervisor first falsifies the current triple and enters the search state. The triple with the next highest score in (3) will be selected for the consensus check. This search process can be repeated from a high score triple to a low score triple until the consensus is

achieved and the control is switched to the tracking state. Assume that in the sampling period of a frame, the computer can process 5 triples' consensus checks, including solving P3P and evaluating matching errors. The experiment for cube recognition and tracking in Fig.2 is carried out again by using the RANSAC method with n=6 and ε =20 pixels. The tracking errors are shown in Fig.5.a, with the dashed line to indicate the searching or tracking state of control. The model is illustrated as the black skeleton of the cube and its tracking trajectory can be observed from Fig.5.b to Fig.5.e.

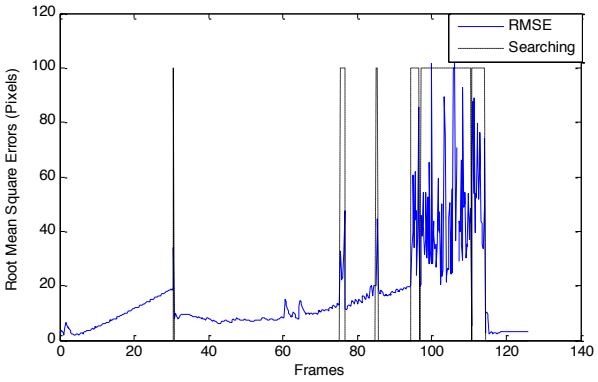


Fig.5.a

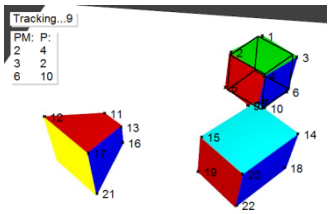


Fig.5.b

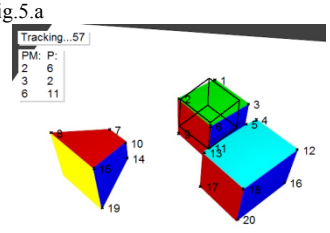


Fig.5.c

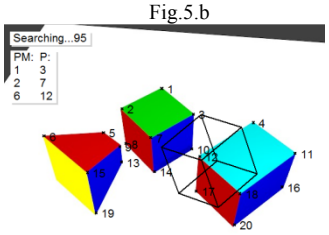


Fig.5.d

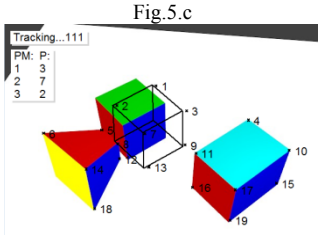


Fig.5.e

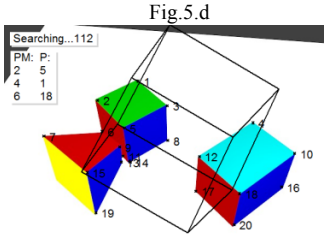


Fig.5.f

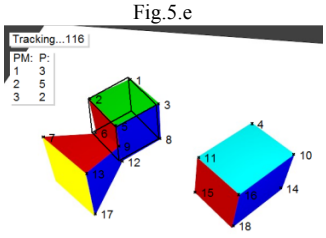


Fig.5.g

Fig.5. RANSAC based relocalization for visual servoing, see video Cube_RANSAC.mov. (a) Tracking error. (b) Tracking at frame 9. (c) Tracking at frame 57. (d) Feature search at frame 95. (e) Tracking at frame 111. (f) Feature search at frame 112. (g) Tracking at frame 116.

Similar to visual servoing in A.1, the cube was accurately tracked until the vertex 6 on the model began to be occluded by the cuboid and was wrongly matched with point 10, as shown in Fig.5.b. Such a spurious match due to the occlusion of the cuboid caused big tracking errors in the following frames, as shown in Fig.5.c, but was within the tolerance for the consensus check. In this experiment, it was of interest to see if tracking could recover from the failure due to the incorrect matching after the cube moved out from the cuboid's occlusion. As shown in Fig.5.d, vertices 1 and 2 were matched correctly but vertex 6 matched with point 12 wrongly. Due to the tracking error e being too big with $RMSE(e) > \varepsilon$, the supervisor started searching for triples with poses that can pass the consensus check. The consensus check was carried out in descending order of the similarity scores of triples. Unfortunately, it took 73 trials from the 95th frame in order to identify the matched pose. As assumed before, 5 trials can be processed for consensus check in each servoing period. It had been frame 111 when the matched triple, $f_1^M \leftrightarrow f_3$, $f_2^M \leftrightarrow f_7$, $f_3^M \leftrightarrow f_2$, was found, as shown in Fig.5.e. However, due to the moving of the cube, it had moved to a new pose with a big difference, i.e. $RMSE(e) > \varepsilon$ in Fig.5.e and the control switched to searching state again in Fig.5.f. Tracking successfully recovered from frame 116 in Fig.5.g after the cube reduced its speed and stopped at frame 119. It can be concluded from this experiment that RANSAC based relocalization is suitable for a static or slow moving object but may fail for a fast moving object.

A.3: Unfalsified Visual Servoing

The proposed unfalsified visual servoing introduces a supervisor on top of visual servo, which monitors both feature similarity and spatial context information for feature selection in every servo cycle.

As is the case with other methods, the control has two states: 1) tracking and 2) searching, where state switching is controlled by the supervisor according to tracking performance. Step 1) in the algorithm corresponds to the searching state. It is in fact a recognition process that identifies feature points from an image according to the model of the object to be recognized. If there are enough consensus or unfalsified features, the state is switched to tracking, as in step 7). The tracking takes a local search approach as in other two methods to improve robustness and reduce searching space for real-time tracking, i.e. finding the matched features $p(\xi^*)$ in the next frame i+1 locally around the predicted projection from the previous object pose in the i^{th} frame, as in step 2).

In the control algorithm, a 3D object can be recognized if at least 6 features can be matched with the model, i.e. n=6. The candidate radius is assumed to be $\varepsilon_1 = 20$ pixels. The consensus radius is assumed to be 2 pixels and the deadzone for ε-minimization is $\varepsilon = 0.1$. The control gain is set to K=1.1. The control errors are shown in Fig.6.a, which demonstrate much better tracking performance than the conventional methods in Fig.4.a and Fig.5.a. For example, it can track the cube accurately before both vertex 4 and vertex 5 were occluded by the cuboid at frame 31. However both similarity based and RANSAC based feature correspondences result in a spurious match but the controllers were unaware of this bad match because the resulted consensus errors are still within the tolerance. The supervisor of the unfalsified visual servo selects the 6 best performing feature points as unfalsified features for tracking according to their tracking performance I (i,k) in step 4) of the algorithm, as shown in Fig.6.b, where i indicates the vertex on the model and k represents the k^{th} sampling frame. The evaluation of tracking performance provides adaptive

capability of dynamic feature selection, for example the spurious matches corresponding to vertex 6 and 8 on the model were excluded from the feedback loop in Fig.7.b.

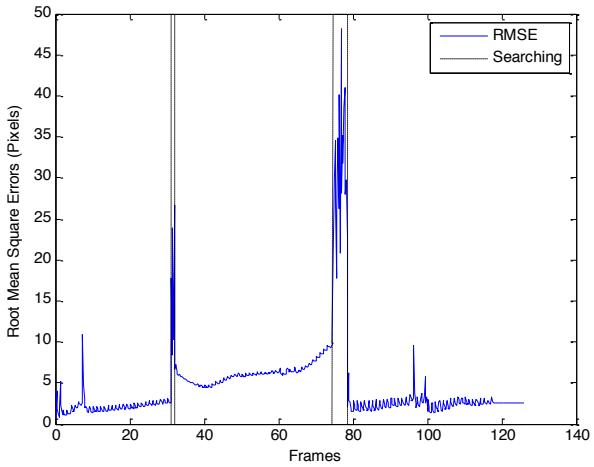


Fig.6.a

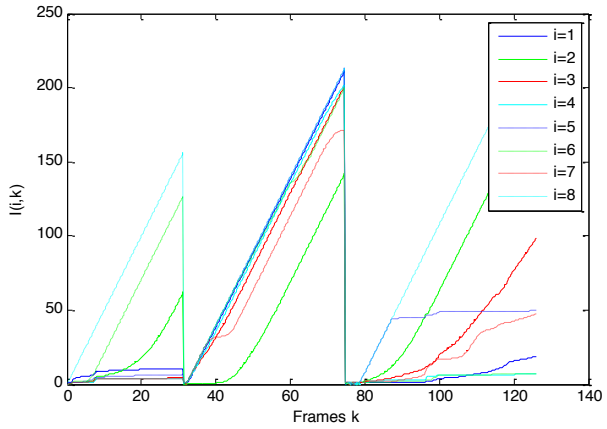


Fig.6.b

Fig.6. Unfalsified visual servoing control. (a) Tracking error. (b) Performance indices of 8 feature points

From Fig.6.b, vertex 1 (solid blue) and vertex 8 (dotted cyan) were not selected by the supervisor due to their poor tracking performance at the beginning of tracking. A snapshot of initial tracking was shown in Fig.7.a, where the unfalsified set was shown as PM and the correspondences in the image were shown as P. As the cube moved, vertex 6 (point 8 in Fig.7.a) started to be occluded by the cuboid and its performance index increase significantly (dotted green in Fig.6.b). Through the ε-cost minimization, vertex 6 was excluded from the unfalsified set but vertex 1 (solid blue in Fig.6.b) was considered to be unfalsified for the visual servoing, as shown in Fig.7.b at frame 9. Because the correct correspondences were achieved, the cube was tracked well by the visual servoing until vertex 5 (point 6 in Fig.7.c) was occluded by the cuboid too, at which point the control was switched to the search state in Fig.7.c. However there were only 5 vertices visible, namely vertices 1,2,3,4 and 7 on the model. With n=6 in this experiment, correct correspondences were not achievable and spurious matches had to be used in the further visual tracking, as shown in Fig.7.d. Whilst vertex 6 was moving out from occlusion the spurious matches, e.g. $f_6^M \leftrightarrow f_{11}$ in Fig.7.d, caused big control error and initiated the search state at frame 75 as shown in Fig.7.e. Different from the RANSAC approach, where the consensus check is carried out off-line for a sampled image and may lose the tracking of a fast moving object, the unfalsified visual servoing integrates the consensus check into its local tracking and therefore the supervisor checks the consensus for each sampled image and then track those detected interest points in the next frame. As assumed before, the supervisor can check the consensus for 5 triples in each servo period. This means the tracking can be quickly recovered from the current frame once a triple passes the consensus check at the step 7) of the algorithm, as shown in Fig.7.f. The cube was successfully tracked with correctly matched feature points just after 6 features on the cube became visible.

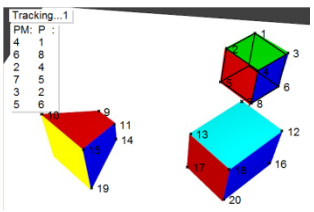


Fig.7.a

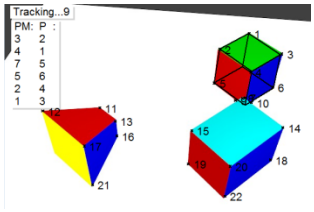


Fig.7.b

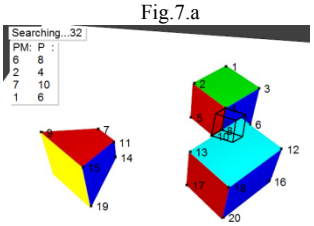


Fig.7.c

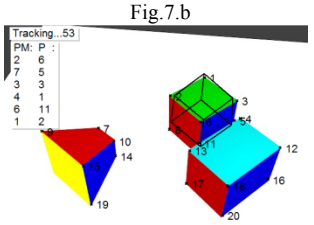


Fig.7.d

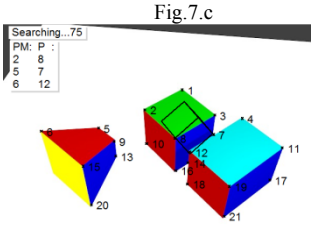


Fig.7.e

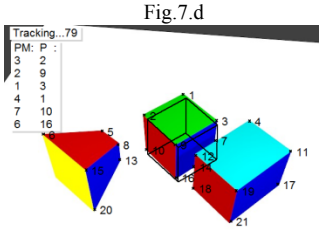


Fig.7.f

Fig.7. Unfalsified visual servoing control, see video Cube_UVS.mov. (a) Tracking at frame 1. (b) Tracking at frame 9. (c) Feature searching at frame 32. (d) Tracking at frame 53. (e) Feature searching at frame 75. (f) Tracking at frame 79.

In comparison with the previous two conventional methods, three main advantages of the unfalsified visual servoing can be observed. First, unfalsified visual servoing has adaptive feature selection capability, which makes it robust to poor features due to occlusions, illumination change, or fast movement. Second, it provides high-level failure detection and recovery to the low-level servo, which makes it more suitable for real-time applications with moving objects. Third, since visual servoing is carried out by feedback of all unfalsified features, higher tracking accuracy can be expected. This can also be illustrated by the tracking performances of the three methods listed in Table 1. Conventional visual servoing using maximum similarity for feature matching shows the highest tracking error due to mismatching as discussed earlier. The proposed unfalsified visual servoing can achieve the best tracking accuracy with a mean tracking error of 3.69 pixels. The main error stems from the mismatch when the cuboid occluded 2 vertices of the cube. The Tracking Percentage represents the

percentage of the control in the state of tracking, in comparison with the searching state for relocalization through global trials. Higher tracking percentage means higher efficiency of recovery from failures for an algorithm, meaning that the unfalsified visual servoing performs best out of the three methods considered. The Relocalization Count shows how many times the control detects a failure and switches into the global search state. It demonstrates that the RANSAC was in the relocalization state 6 times in comparison with the 2 times seen for the other methods, a consequence of its slow off-line computational time. It causes the calculated pose to lose the tracking due to the motion of the object and has to be relocalized again. From this table, we can conclude that the proposed unfalsified visual servoing outperforms the other two.

TABLE I
TRACKING PERFORMANCES OF CONVENTIONAL VISUAL SERVOING, RANSAC BASED VISUAL SERVOING, UNFALSIFIED VISUAL SERVOING

Control Method	Mean Tracking Error(Pixels)	Tracking Percentage(%)	Relocalization Count
Conventional VS	10.75	65.08	2
RANSAC VS	9.62	82.70	6
Unfalsified VS	3.69	96.03	2

B. Teabox Tracking under Adverse Conditions

In this section, real visual tracking of a tea box was implemented. At first, a 3D model of the tea box of $140mm\times 85mm\times 45mm$ was built. Interest points on the surface were detected by the binomial approximation of the Harris operator. Finding correspondence between the interest points on the model and those in an image was considered as a Bayesian classification problem for fast implementation [31, 46], where randomized ferns were used with a fern size of 11 and 30 ferns altogether. The randomized ferns were trained off-line by using a sampled image of the tea box, which took 110 seconds for a patch size of 30 pixels. After the ferns were trained, it was ready for feature points matching in real-time, which took an average of 12 milliseconds in the experiments. The fast matching capability of randomized ferns enables real-time failure recovery of the proposed algorithm.

The camera was calibrated in Halcon and the intrinsic parameters were obtained in Table II:

TABLE II INTRINSIC CAMERA PARAMETERS

Focal length Foc(m)	0.0531860
Radial distortion coefficient Kappa ($1/m^2$)	30.3800072
Width/Height of a cell on the sensor [Sx,Sy] (m)	[3.4898646e-005, 3.1042e-005]
X and Y-coordinates of the image center [Cx, Cy] (pixels)	[310.1272562, 239.8797951]

During visual tracking, an image is sampled and the trained tea box is recognized and tracked. The unfalsified visual servoing was implemented with control parameters in Table III.

TABLE III CONTROL PARAMETERS

n minimum number of feature points	10
d_t for tube-collinearity test	22.36 pixels
ε for ε-minimization	0.1
Consensus radius ε_2	2 pixels
Candidate radius ε_1	20 pixels
Control gain K	0.5

B.1: Tracking the Tea Box

Tracking of the tea box in a cluttered background was experimented as shown in Fig.8. The motion of the tea box can make some features invisible or even out of the camera's field-of-view, as seen bottom-right in Fig.8. The experiment demonstrated that the proposed unfalsified control can dynamically select well-performed features for tracking and can automatically search for the object to recover from tracking failure. The root mean square (RMS) of image errors from the 1st frame to the 1175th frames are shown in Fig.9.a. It achieved an average RMS error of 1.3202 pixels for the whole tracking. The spikes in Fig.9.b illustrate the switching from local tracking to recognition controlled by the supervisor, when a global search is activated after too few feature points can be matched or detected in the image. For example, from frame 990 to frame 1080, the tea box was moved out of the field-of-view of the camera. The algorithm kept matching feature points with the model continuously until the tea box appeared again, i.e. when more than 10 feature points were detected in an image. The algorithm was in tracking state afterwards.

As a comparison, the scheme used in section A2, considering both similarity scoring and space indexing[32], was implemented with control gain K=1, where tracking is failed and global search is initiated if the RMS of image errors exceed a threshold (20 pixels) or too few features can be matched in an image (10 points). The tracking errors are illustrated in Fig.9.c. The average RMS of image errors reaches 2.93 pixels. The poor tracking can be due to mismatched features used in the feedback that cause transient behaviours such as jittering. With the unfalsified visual servoing, tracking performances of feature points are observed. Only the best performed features are selected for the feedback. If all features perform poorly, the current pose is falsified and tracking stops, which can be restarted only if enough interest points similar to the modelled features are detected. This makes the proposed supervisory control safer for robot visual servo applications in an unstructured environment because wrong features used in the feedback loop may result in severe performance degradation.

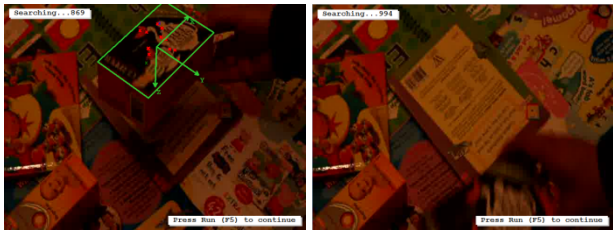


Fig. 8. Tracking the tea box. The green coordinates system is the detected pose and the green rectangle is the top face of the tea box. The unfalsified interested points used in the visual servoing are shown as red crosses.

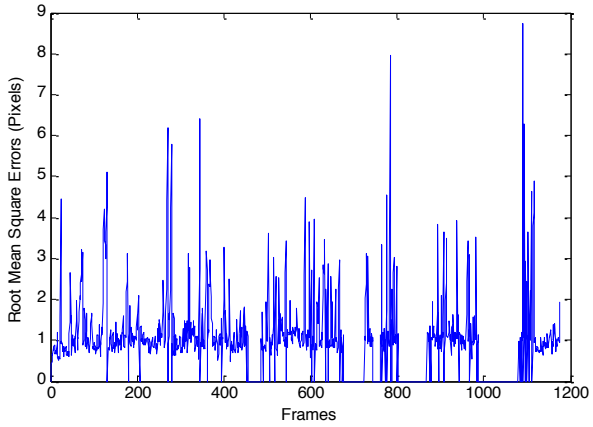


Fig.9.a

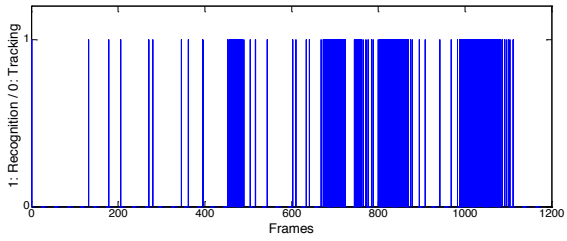


Fig.9.b

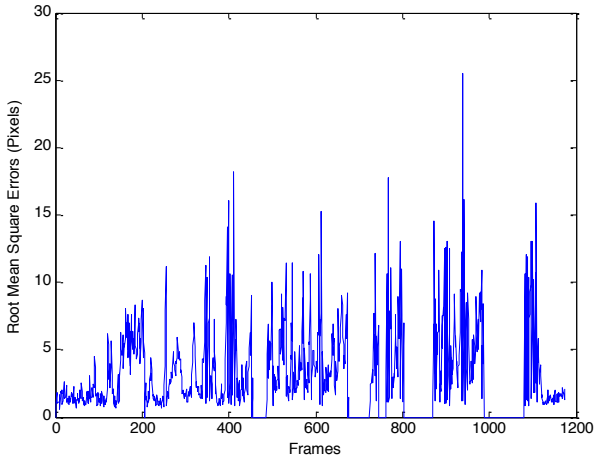


Fig.9.c

Fig. 9 . (a) Root mean square errors of the unfalsified visual servoing. (b) Events of global search for recovery from matching failure. (c) Root mean square errors of conventional visual servoing with RANSAC relocalization.

B.2: Dealing with Occlusions

Occlusions often cause matching failure, which can seriously affect visual servoing performance. The proposed unfalsified visual servoing provides a mechanism to deal with occlusions under the supervisory control paradigm, where a supervisor monitors the tracking performance of feature points and selects visible features dynamically for control feedback. If complete occlusion occurs, the supervisor starts a global search in order to reinitialize tracking, which corresponds to the recognition mode of the algorithm. Fig. 10 showed a tracking example of the unfalsified visual servoing from frame 1287 to 1776, where the tea box was covered by a book several times as shown in Fig.10.a. From the red crosses in Fig.10.a, which indicate the selected feature points for visual servo, we can see that visible feature points were dynamically updated and selected into the unfalsified set for visual servo during partial occlusion, for example from frame 1370 to 1396. After frame 1396, the book covered the tea box completely. Less than 10 unfalsified feature points can be detected in the image; global search was repeatedly activated in order to identify the tea box from the images as shown in Fig.10.c. The tracking errors of the whole process are shown in Fig.10.b, with an average RMS error of 1.0225 pixels during tracking.

Fig.10.a

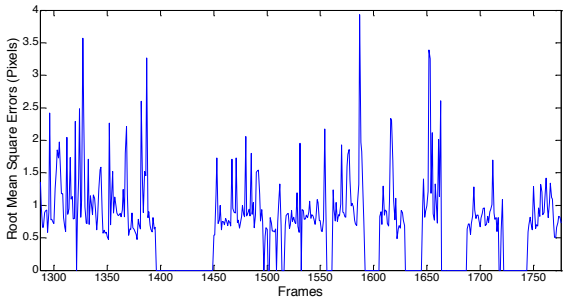


Fig.10.b

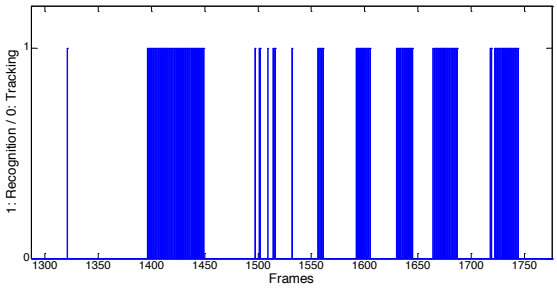


Fig.10.c

Fig. 10. (a) A book was moved above the tea box 6 times. Detected feature points were dynamically selected into unfalsified set for control. (b) Root mean square errors of the unfalsified visual servoing. (c) Events of global search for recovery from matching failure.

B3: Dealing with Illumination Change

Another main uncertainty that may cause feature point mismatching is illumination variation, which is common in most uncontrolled environments. Fig. 11 shows an example of tracking under variable illumination, where the brightness of a light was changed through a dimmer three times and shadow was also introduced from frame 1885 to 2640 as shown in bottom-right of Fig.11.a. Several example images are shown in Fig.11.a. The tracking errors of the unfalsified visual servoing are shown in Fig.11.b, with an average error of 1.1795 pixels. In Fig.11.c three wider bars showed that the brightness was so low that the modelled feature points cannot be detected in the images, for example from frame 2020 to 2130. As a result, the global search was initialized and the tracking restarted when features were detected in the images again.

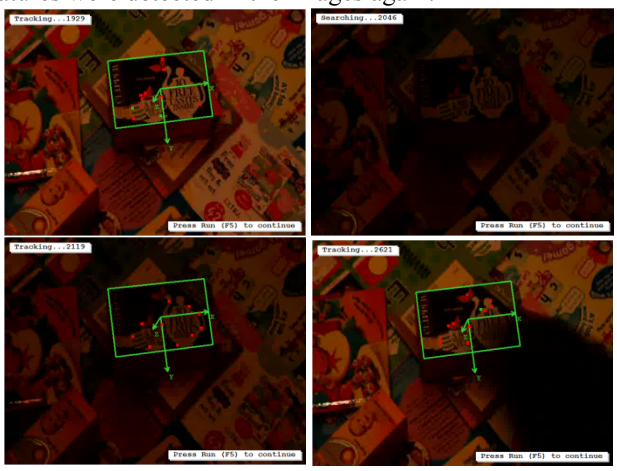


Fig.11.a

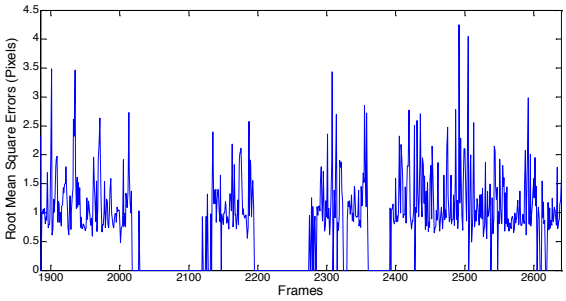


Fig.11.b

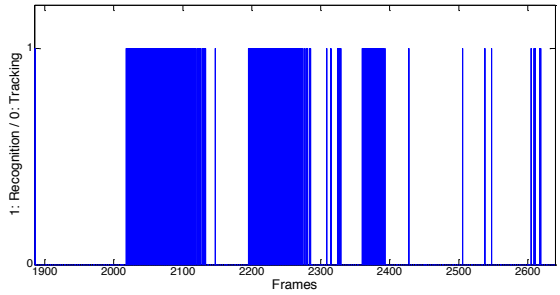


Fig.11.c

Fig. 11. (a) Brightness of a light was changed during the tracking. (b) Root mean square errors of the unfalsified visual servoing. (c) Events of global search for recovery from matching failure.

The above experiments demonstrated the capability of feature selection for visual tracking in real environments. Even though the tracked features could be missed or lost due to ambiguity or disturbance, the unfalsified visual servoing can automatically switch between global search and local tracking and recover from failures without human intervention. The video of experiments *B.1*, *B.2* and *B.3* can be found in Teabox_UVS.mov.

VI. Conclusion

Due to various uncertainties existing in a real environment, visual tracking is inherently fragile and requires a fault tolerance design in order to recover from any failure automatically. An unfalsified adaptive controller for visual tracking of a 3D object has been proposed to deal with unreliable features extracted from an image. The proposed controller includes locally stable tracking control supervised by a switching mechanism for feature selection and global relocalization for recovery from tracking failure. The extracted features in the image can be falsified or unfalsified for tracking control by evaluation of their tracking history. Since the unfalsified control is completely data-driven, it can switch features in or out the control loop dynamically for reliable feature selection or fault rectification. From a cognitive point of view, the proposed algorithm provides a low-level servo controller with some intelligent aspects, such as visual attention and context-awareness. The proposed algorithm was implemented for visual tracking of modelled objects under adverse conditions, such as fast motion, cluttered background, occlusions, illumination variation and movement out of the field-of-view. Recognition and tracking process coordinated by the supervisor were demonstrated and satisfied tracking performance was achieved.

References

[1] E. Marchand and F. Chaumette, "Feature tracking for visual servoing purposes," *Robotics and Autonomous Systems,* vol. 52, pp. 53-70, 2005.

[2] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation,* vol. 12, pp. 651-670, 1996.

[3] A. I. Comport, E. Marchand, M. Pressigout, and F. Chaumette, "Real-time markerless tracking for augmented reality: the virtual visual servoing framework," *IEEE Transactions on Visualization and Computer Graphics,* vol. 12, pp. 615-628, 2006.

[4] T. Okuma, T. Kurata, and K. Sakaue, "A natural feature-based 3D object tracking method for wearable augmented reality," presented at the The 8th IEEE International Workshop on Advanced Motion Control Kawasaki, Japan, 2004.

[5] F. Janabi-Sharifi, L. Deng, and W. J. Wilson, "Comparison of basic visual servoing methods," *IEEE/ASME Trans on Mechatronics,* vol. 16, pp. 967-983, 2011.

[6] V. Teichrieb, J. P. S. d. M. Lima, E. L. Apolinario, T. S. M. C. d. Farias, M. A. S. Bueno, J. Kelner, and I. H. F. Santos, "A Survey of Online Monocular Markerless Augmented Reality," *International Journal of Modeling and Simulation for the Petroleum Industry,* vol. 1, pp. 1-7, 2007.

[7] X. Gratal, J. Romero, J. Bohg, and D. Kragic, "Visual servoing on unknown objects," *Mechatronics,* vol. 22, pp. 423–435, 2012.

[8] A. Davison, I. Reid, N. Molton, and O. Stasse, "MonoSLAM: Real-Time Single Camera SLAM," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 29 pp. 1052-1067, 2007.

[9] V. Lepetit and P. Fua, "Monocular model-based 3D tracking of rigid objects: a survey," *Foundations and Trends in Computer Graphics and Vision,* vol. 1, pp. 1–89, 2005.

[10] E. Malis, "Stability analysis of invariant visual servoing and robustness to parametric uncertainties," in *Control Problems in Robotics* vol. 4, Antonio Bicchi, Domenico Prattichizzo, and H. I. Christensen, Eds., ed: Springer, 2003, pp. 265-280.

[11] Z. Dong, G. Zhang, J. Jia, and H. Bao, "Efficient keyframe-based real-time camera tracking," *Computer Vision and Image Understanding,* vol. 118, pp. 97-110, 2014.

[12] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applicatlons to image analysis and automated cartography," *Communications of the ACM 24 (6): 381–395,* vol. 24, pp. 381–395, 1981.

[13] L. Vacchetti, V. Lepetit, and P. Fua, "Stable real-time 3D tracking using online and offline information," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 26, pp. 1385–1391, 2004.

[14] M. G. Safonov and T.-C. Tsao, "The unfalsified control concept and learning," *IEEE Trans. on Automation and Control,* vol. 42, pp. 843-847, 1997.

[15] P. Jiang and R. Unbehauen, "Robot visual servoing with iterative learning control," *IEEE Trans. Syst.,Man, Cybern. A, Syst., Humans,* vol. 32, pp. 281-287, 2002.

[16] F. Conticelli and B. Allotta, "Discrete-time robot visual feedback in 3-D positioning tasks with depth adaptation," *IEEE/ASME Trans. Mechatronics,* vol. 6, pp. 356–363, 2001.

[17] E. Malis, Y. Mezouar, and P. Rives, "Robustness of image-based visual servoing with a calibrated camera in the presence of uncertainties in the three-dimensional structure," *IEEE Trans on Robotics,* vol. 26, pp. 112-120, 2010.

[18] J. A. Piepmeier, G. V. McMurray, and H. Lipkin, "Uncalibrated dynamic visual servoing," *IEEE Trans. Robot. Autom.,* vol. 20, pp. 143–147, 2004.

[19] P. Jiang, L. Bamforth, Z. Feng, J. Baruch, and Y. Chen, "Indirect iterative learning control for discrete visual servo without a camera-robot model," *IEEE Trans. on System, Man, and Cybernetics-Part B,* vol. 37, pp. 863-876, 2007.

[20] M. Hao and Z. Sun, "A universal state-space approach to uncalibrated model-free visual servoing," *IEEE/ASME Trans on Mechatronics,* vol. 17, pp. 833-846, 2012.

[21] P. Jiang, H. Chen, and L. Bamforth, "A universal iterative learning stabilizer for a class of MIMO systems," *Automatica,* vol. 42, pp. 973-981, 2006.

[22] J. Su, "Convergence analysis for the uncalibrated robotic hand-eye coordination based on the unmodeled dynamics observer," *Robotica,* vol. 28, pp. 597-605, 2010 2010.

[23] N. R. Gans and S. A. Hutchinson, "Stable visual servoing through hybrid switched-system control," *IEEE Trans on Robotics,* vol. 23, pp. 530-540, 2007.

[24] D.-H. Park, J.-H. Kwon, and I.-J. Ha, "Novel position-based visual servoing approach to robust global stability under field-of-view constraint," *IEEE Trans on Industrial Electronics,* vol. 59, pp. 4735-4752, 2012.

[25] N. García-Aracil, E. Malis, R. Aracil-Santonja, and C. Pérez-Vidal, "Continuous visual servoing despite the changes of visibility in image features," *IEEE Trans on Robotics,* vol. 21, pp. 1214-1220, 2005.

[26] A. Assa and F. Janabi-Sharifi, "A robust vision-based sensor fusion approach for real-time pose estimation," *IEEE Trans on Cybernetics,* vol. 44, pp. 217-227, 2014.

[27] B. Williams, G. Klein, and I. Reid, "Automatic relocalization and loop closing for real-time monocular SLAM," *IEEE Trans on Pattern Analysis and Machine Intelligence,* vol. 33, pp. 1699-1712, 2011.

[28] G. Klein and D. Murray, "Parallel tracking and mapping on a camera phone," presented at the International Symposium on Mixed and Augmented Reality (ISMAR'09), Orlando, 2009.

[29] R. Spica, P. R. Giordano, and F. Chaumette, "Active structure from motion: application to point, sphere, and cylinder," *IEEE Trans. on Robotics,* vol. 30, pp. 1499-1513, 2014.

[30] Y. Xie, W. Zhang, C. Li, S. Lin, Y. Qu, and Y. Zhang, "Discriminative object tracking via sparse representation and online dictionary learning," *IEEE Trans on Cybernetics,* vol. 44, pp. 539-551, 2014.

[31] V. Lepetit and P. Fua, "Keypoint recognition using randomized trees," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, pp. 1465-1479, 2006.

[32] D. Chekhlov, W. Mayol-Cuevas, and A. Calway, "Appearance based indexing for relocalisation in real-time visual SLAM," presented at the 19th Bristish Machine Vision Conference, Leeds, UK, 2008.

[33] J. Straub, S. Hilsenbeck, G. Schroth, R. Huitl, A. Moller, and E. Steinbach, "Fast relocalization for visual odometry using binary features," presented at the 20th IEEE International Conference on Image Processing Melbourne, VIC, 2013.

[34] M. Calonder, V. Lepetit, and P. Fua, "BRIEF : Binary Robust Independent Elementary Features," presented at the 11th European Conference on Computer Vision, Crete, Greece 2010.

[35] D. G. Lowe, "Object recognition from local scale invariant features," presented at the International Conference on Computer Vision, Corfu, Greece, 1999.

[36] S. Hinterstoisser, S. Benhimane, and N. Navab, "N3M: natural 3D markers for real-time object detection and pose estimation," presented at the IEEE 11th International Conference on Computer Vision Rio de Janeiro, 2007.

[37] A. S. Morse, "Supervisory control of families of linear set-point controllers-part 1 : exact matching," *IEEE Trans on Automatic Control,* vol. 41, pp. 1413-1431, 1996.

[38] J. Hespanha, "Tutorial on supervisory control," presented at the the 40th Conf. on Decision and Control, Orlando, FL, 2001.

[39] T. C. Tsao and M. Safonov, "Unfalsified direct adaptive control of a two-link robot arm," *International Journal of Adaptive Control and Signal Processing,* vol. 15, pp. 319–334, 2001.

[40] R. Wang, A. Paul, M. Stefanovic, and M. G. Safonov, "Cost detectability and stability of adaptive control systems," *Int. J. Robust Nonlinear Control,* vol. 17, pp. 549–561, 2007.

[41] J. v. Helvoort, B. d. Jager, and M. Steinbuch, "Direct data-driven recursive controller unfalsification with analytic update," *Automatica,* vol. 43, pp. 2034-2046., 2007.

[42] G. Battistelli, J. Hespanha, E. Mosca, and P. Tesi, "Unfalsified adaptive switching supervisory control of time varying systems," presented at the the 48th IEEE Conference on Decision and Control and the 28th Chinese Control Conference, Shanghai, China, 2009.

[43] A. Ingimundarson and R. S. S. Pena, "Using the unfalsified control concept to achieve fault tolerance," presented at the the IFAC World Congress, Seoul, Korea, 2008.

[44] M. Stefanovic, R. Wang, and M. G. Safonov, "Stability and convergence in adaptive systems," presented at the American Control Conference, Boston, MA, 2004.

[45] A. Paul and M. G. Safonov, "Model reference adaptive control using multiple controllers and switching," presented at the the 42nd IEEE Conference on Decision and Control, Maui, Hawaii USA, 2003.

[46] M. Ozuysal, P. Fua, and V. Lepetit, "Fast keypoint recognition in ten lines of code " in *IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, MN, USA, 2007.