MVA2007 IAPR Conference on Machine Vision Applications, May 16-18, 2007, Tokyo, JAPAN

Tracking Features With Global Motion Compensation for Drone Camera Servoing

Benoit Louvat, Laurent Bonnaud, Nicolas Marchand and Gerard Bouvier
GIPSA-lab INPG/ENSIEG, Grenoble, France
benoit.louvat@lis.inpg.fr, laurent.bonnaud@inpg.fr, nicolas.marchand@lag.ensieg.inpg.fr, gerard.bouvier@inpg.fr

Abstract

This paper deals with visual servoing for a pan and tilt camera embedded in a drone. Video is transmitted to the ground where images are processed on a PC, and turret controls are sent back to the drone. The objective is to track any fixed object on the ground without knowledge about shape or texture and to keep it centered in the image. In order to achieve this task an algorithm that combines feature-based and global motion estimation is proposed. This algorithm provides a good robustness to very strong video transmission noise and works at a frame rate close to 25 fps. The control of the system is based on a double closed loop, which achieves a fast convergence to the desired position. Experimentation in real conditions shows the effectiveness of the proposed scheme.

1 Introduction

The most critical part in a visual servoing system is to find reliable visual information to estimate the motion between two images. These algorithms should be robust to noise, especially in our case where a strong noise due to the video transmission can appear. They should be accurate and provide estimation of large displacement due the drone's motion. They should also have a low computational cost in order to achieve a fast processing rate. In that field, there are lot of studies. The literature can be classified into three classes of contribution. The first class is based on the extraction and tracking of geometric features. The second class relies on the definition of a model for the object to track while the third class uses motion analysis. The first class, known as feature-based focuses on tracking 2D features such as geometrical primitives (points [1, 2], lines, segments, ellipses [3], contours [4]) to deduce the inter-frame displacement. The main advantages of these algorithms are their simplicity and their computational cost. On the other hand, the quality of results depends on features' density, furthermore these algorithms are very sensitive to noise and large displacements. The second class, known as model-based, defines a model of the desired tracked object, such as CAD model [5, 6]. They generally amount to pose estimation. Then the visual servoing task consists in moving the robot until the current pose of the object corresponds to the desired pose. They are more robust but need *a priori* knowledge of the target, which is not realistic in the application context of this paper. The third class of methods is used when the scene is too complex and the extraction of simple primitives is impossible [7, 8, 9]. The approach are here based on the estimation of a set of parameters which describes the transformations and the displacements of a part of the image or of the whole image by minimizing an error function. No extraction of geometrical primitives is needed. The choice of a well suited error function and the use of efficient techniques of minimization allow the description of complex 2D transformations such as affine or homographic motion. The main disadvantages of these methods are their slowness and their lack of accuracy when the target is small compared to the whole image. The first class is very interesting to the visual servoing task of this paper. Indeed it provides tracking of objects contained in a small part of the image with a high degree of accuracy. But due to the video transmission, the quality of images is not constant. Moreover a bad video transmission can severely damage or remove some part of the image. If the damaged part of image contains the tracked object all features are lost and tracking fails. Furthermore large displacements can also cause tracking failures. Consequently, in this context feature-based algorithms are not robust enough. To solve this problem without loosing in accuracy, we propose to combine a feature-based algorithm with the third class of methods. This class provides an estimation of the global motion robust to noise and large displacements. Indeed if damage appears in the image this kind of algorithm permits to estimate the global motion in the part of the image which is not damaged and continue to update the position of features.
In this paper, the first section describes the feature-based and global motion estimation algorithm used (KLT [1] and RMRm [9]). Then a presentation of the proposed algorithm for the visual servoing task is reported. The next section presents the control law scheme based on two closed loops. Afterwards, results obtained by this method with real images will be shown. Finally we will conclude and present some future improvements.

2 Target Motion Estimation Algorithm

2.1 Feature-based algorithm

The extraction of features and their tracking is based on the well-known KLT algorithm [1]. The motion between an image I at time t and an image I at time $t+\tau$, can be described

as follows with the hypothesis that illumination is constant:

$$I(p, t+\tau) = I(p - \theta(p, t, \tau), t)$$

By this way an image taken at time $t+\tau$ can be obtained by moving every points of the image taken at time t by the appropriate amount θ. The vector θ represents the displacement of the point p. This point is not a single pixel but the center of an analysis window W. There are many methods to choose a window [10] and many displacement models for this window. According to computing time and our frame rate we choose a small window and simple translation model [2]. The displacement vector can be written as follows:

$$\forall p \epsilon W \quad \theta(p, t) = d(t)$$

where d is the inter-frame displacement. The problem of solving the motion between two frames is then to find the parameter d that minimizes the dissimilarity

$$E = \sum_{p \epsilon W} [I(p, t+\tau) + \bigtriangledown I(p, t+\tau).d - I(p, t)]^2$$

where $\bigtriangledown I(p, t)$ is the spatial image gradient computed at point p. Finally to find the displacement d the above equation is minimized by the Newton-Raphston algorithm with a mutliresolution strategy.

2.2 Global motion estimation algorithm

In order to estimate the global motion in the image, we define a parametric motion model. To compute it we use the RMRm algorithm described in [9]. A Gaussian image pyramid is constructed at each time. Let be θ_t the vector of the motion model parameters at time t. The first estimation consists in minimizing the criterion

$$C(\theta_t) = \sum_p \rho(\nabla I(p, t).\omega_{\theta_t}(p) + I_t(p, t))$$

where point p are all the points in an estimation support (the whole image or a part of image), I is the intensity function ∇I and I_t are the spatial gradient and temporal derivative $\omega_{\theta_t}(p)$ is the velocity vector at point p provided by θ and ρ a robust estimator such as the Turkey's biweight function. This estimator allows us to reject the outliers, i.e., points p whose spatiotemporal gradient does not correspond to the current estimation of θ. Then, a hierarchical and iterative minimization strategy is used.

2.3 Feature-based algorithm with global motion compensation

The inter-frame motion has to be estimated in order to find the position of the desired object in the image. Considering that the ground filmed by our drone is a planar surface, two characteristics have to be taken into consideration: the quality of images and the size of objects. So the algorithm needs a high degree of accuracy and a good robustness. For that, we combine the two algorithms presented above, KLT for the feature points tracking and RMRm for the estimation of the motion model. The algorithm breaks down into several steps. A first step of initialization is necessary. It includes the extraction of feature points from the part of image which contains the object selected by the user and the initialization of the support to estimate the global motion (i.e. the window where the parametric model is calculated). This support is smaller than the whole image for computing time reasons. On the other hand, it must be large enough in order to contain a sufficient number of pixels even in case of partial image transmission. In practice, estimation support is a window of 200x200 for an image of 320x240. This size also makes possible to limit the effect of edges. The second step which corresponds to the tracking itself is divided into two parts. The first part is the global motion estimation using RMRm. In [11] a quadratic model is advocated to describe the motion. Actually estimating a quadratic model is too long in computational time to be integrated in a closed control loop. Consequently we choose to consider an affine model to describe the global motion. In fact, the target is fixed, so the global motion can be roughly approximated to the target's motion which allows a prediction of the position of feature points.

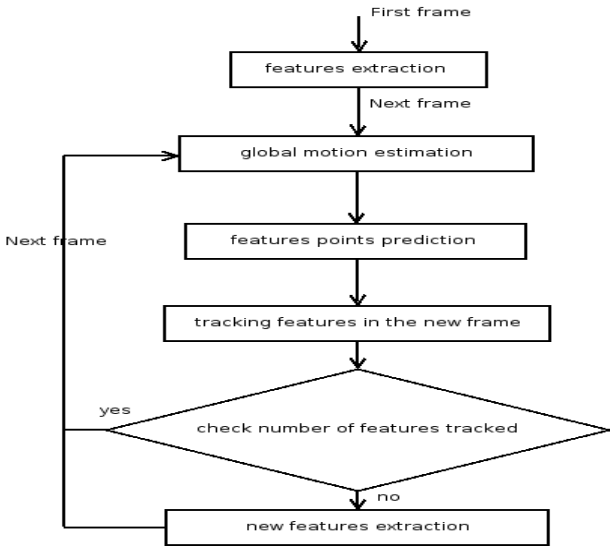

Figure 1: Feature-based tracking with global motion compensation

The second part is the tracking of feature points by using KLT. A multiresolution strategy is used to solve this equation. A pyramid of images is built, the original image is divided into several images with different levels of resolution. Each level represents the original image with a resolution divided by 2. The initialization is performed at the lower level. The value of the displacement d of the feature is initialized by the amount of the global motion previously estimated. All the feature points are moved by this amount that allows refining the research of the target. This process can be assimilated to a feature points prediction. When the minimization is done at this level, the position of the feature point is projected to the upper level and the same scheme is done until the upper resolution level is reached. As we will see in section 4, the great advantage of this initialization by RMRm is the increase in robustness.

Finally, a last step consists in testing the relevance of the feature points found before updating the control. All the

tracked feature points are tested. If the position of a feature is too far from the center of gravity of the group than a threshold, the feature is rejected and the target's position is recomputed. During the second step, it happens that KLT loses feature points on account of the unconvergence of Newton-Raphston algorithm. In order to solve this problem we fix a threshold which corresponds to the minimum number of feature points to continue the tracking. When this threshold is crossed, new feature points are extracted around the target position from the current image. This process allows tracking target for a longer time. Finally, the adjusted position is sent to the control. The algorithm is summarized on figure 1.

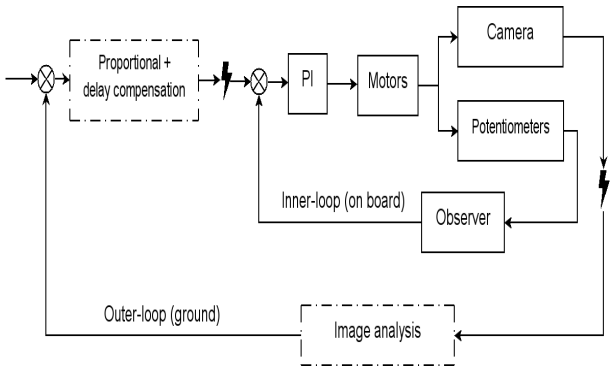

Figure 2: Control loops

3 Control With Two Closed Loops

In visual servoing tasks, the control law scheme is often a simple proportional gain [12]. The dynamics of actuators enabling the pan and tilt movements are neglected and their transfer function are assumed to be a simple gain. In the ideal case (absence of delays, quantization phenomena, friction, etc.), this provides an exponential convergence to the desired position in the image. In the present context, the motion of the target in the image can be fast and neglecting the actuators dynamics is not suitable. Hence, inner control loops were designed to improve the angular speed of the turret. Figure 2 illustrates the control structure. Note that, for simplicity, the pan and tilt inner loops are schematized as a single control loop. Proportional Integral controllers (PI) provide this angular speed control. The PIs are tuned to have a closed loop time constant of 10ms. Since only the angular positions are measured thanks to potentiometers, observers were developed in order to estimate the current angular speed in pan and tilt. This is implemented on chip embedded in the drone. The outer control loop is then designed using the visual information and provides the setting points to the inner loops using the error between the desired (centered) and the current position of the target in the image. A simple proportional gain is not sufficient to insure convergence because of the delays in the loop due to electronics, transmissions and image processing. A prediction term based on a Pade approximation of the delay is added to remove the oscillations introduced by the delays. Note that contrary to classical automatic control theory, delays and time periods are linked to the computational cost of the image and control tasks and hence are not constant. These variations are explicitly taken into account in the control law.

4 Results

All tests have been made with real images from the drone. During a fly, tests are not reproducible. Due to this, different algorithms are not comparable using the visual servoing task. Furthermore real sequences do not authorize to test a large range of object's displacement. For all these reasons, tests are made with a synthetic motion. This motion is the displacement of a house between two images as shown on figure 3. The first validation step demonstrates the usefulness of RMRm in the algorithm proposed by comparing KLT alone and KLT with RMRm. During a flight interframe's displacement of objects vary from small to large, due to the drone motion. According to this we create a house's motion with amplitudes varying from 1 pixel to 65 pixels. Figure 4 shows the motion estimation error with respect of the displacement amplitude of the house. Four features are extracted and tracked. We can see that KLT with global motion estimation improves results compared to KLT alone. The error is always equal to zero whereas KLT gives a bad estimation when the displacement is larger than 20 or 40 pixels. Figure 5 shows that the number of features lost during the tracking is more important when the initialization of KLT is not performed by global motion. As we have seen in section 3, KLT serves to improve the accuracy of the motion estimation. To demonstrate the usefulness of KLT, we trace on the figure 6 the motion estimated by RMRm and the motion estimated by our algorithm with respect to RMRm's iterations for a displacement of 30 pixels of the house with four features extracted. It can be noticed that KLT improves the global motion estimated by RMRm. This improvement appears for a few iteration of RMRm. This characteristic can be used to set the maximum number of iterations for the two algorithms in order to accelerate the visual servoing task without loosing information. In conclusion we can say that each algorithm corrects errors of the other one.
In order to completely validate the approach we test the proposed algorithm embedded in the visual servoing closed loop described section 3. On the film taken during a flight [13], it can be seen that the proposed algorithm is robust to bad video transmission. When the image is damaged features prediction provided by global motion estimation allows the visual control system to continue tracking the target. The processing rate is 20 frames per second with RMRm set to 4 iterations and KLT set to 10 iterations maximum. The PC for vision processing is a Pentium 4 3.2GHz.

5 Conclusion

We have proposed an algorithm to track objects without knowledge on the target. The behavior of this algorithm respects visual control system's constraints describe before. It is accurate enough to track small target even in case of large displacement. It provides a robust tracking to video

transmission problem when part of images are damaged or removed. Finally it is fast enough to be embedded in a visual servoing closed loop. However the robustness to video transmission problem is not flawless. When a large part of the image is damaged global motion estimation can fail and features are lost. Indeed the motion estimation support may contain a majority of damaged pixels which leads to a bad motion estimation. In this case undamaged pixels are likely to be available outside the estimation support. Therefore we could improve the global motion estimation by detecting damaged pixels and removing them from the estimation support. In order to retain enough pixels in the estimation support we could extend the estimation support window to include more undamaged pixels. Further developments may also concern the tracking of mobile objects with respect to the ground.

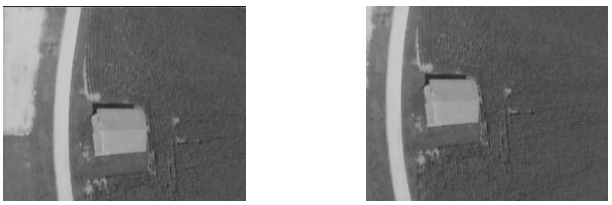

Figure 3: Displacement of 30 pixels of a house

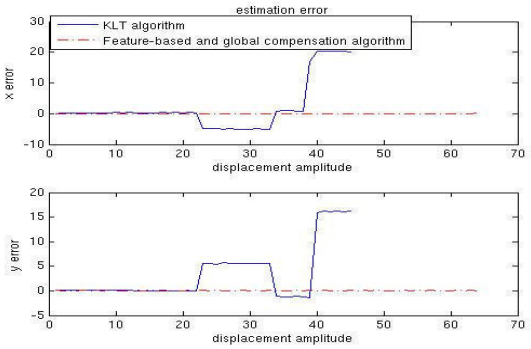

Figure 4: Error with respect to the amplitude of house's displacement

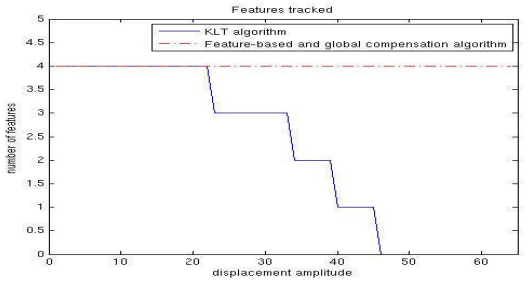

Figure 5: Number of feature points with respect to the amplitude of motion

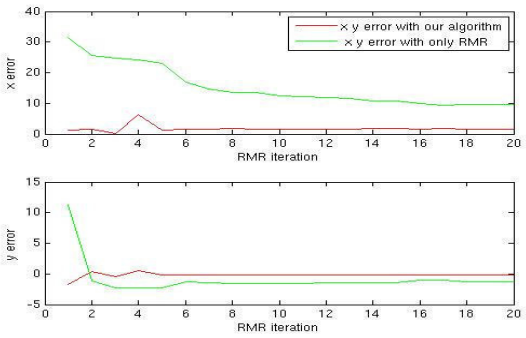

Figure 6: Global motion improved by KLT

References

[1] T. Kanade and C. Tomasi, "Detection and tracking of point features," Tech. Rep. CMU-CS-91-132, Carnegie Mellon University, Apr. 1991.

[2] J. Shi and C. Tomasi, "Good features to track," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR'94*, (Seattle, USA), pp. 593–600, June 1994.

[3] M. Vinsze, "Robust tracking of ellipses at frame rate," *Pattern Recognition*, vol. 34, pp. 487–498, Feb. 2001.

[4] M. Berger, "How to track efficiently piecewise curved contours with a view to reconstructing 3d objects," in *Proc. of the Int. Conf. on Pattern Recognition, ICPR'94*, (Jerusalem, Israel), pp. 32–36, Oct. 1994.

[5] E. Marchand, P. Bouthemy, and F. Chaumette, "A 2D-3D model-based approach to real-time visual tracking," *Image and Vision Computing*, vol. 19, no. 13, 2001.

[6] D. Kragic and H. Christensen, "Model-based techniques for robotics visual servoing and grasping," in *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems, IROS'02*, vol. 1, (Lausanne, Switzerland), pp. 299–304, Oct. 2002.

[7] C.-W. Lin, Y.-J. Chang, and Y.-C. Chen, "Hierarchical motion estimation algorithm based on pyramidal successive elimination," in *Proc. of the Int. Computer Symp.*, (Tainan, Taiwan), pp. 41–44, Oct. 1998.

[8] G. Hager and K. Toyama, "Efficient region tracking with parametric models of geometry and illumination," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 1025–1039, Oct. 1998.

[9] J.-M. Odobez and P. Bouthemy, "Robust multiresolution estimation of parmetric models applied to complex scenes," tech. rep., IRISA, Jan. 1994.

[10] M. Okutomi and T. Kanade, "A locally adaptive window for signal matching," *Int. Journal of Computer Vision*, vol. 7, pp. 143–162, Jan. 1992.

[11] M. Subarrao and A.-M. Waxman, "Closed-form solutions to image flow equations for planar surface in motion," *Computer Vision, Graphics, and Image Processing*, vol. 36, no. 2-3, pp. 208–228, 1986.

[12] A. Cretual and F. Chaumette, "Visual servoing based on image motion," *Int. Journal of Robotics Research.*, vol. 20, no. 11, pp. 857–877, 2001.

[13] "Demonstration video." Internet address: http://film.drone.free.fr/.