The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

A Disaster Invariant Feature for Localization

Behdad Soleimani, Mohammad-Hassan Zokaei Ashtiani, Behrouz Haji Soleimani, Hadi Moradi,
Members, IEEE

Abstract—**In this paper we present a *Disaster Invariant Feature* (DIF), which is used for localization of Unmanned Aerial Vehicles (UAV). There exist numerous researches that address the problem of localization of UAVs using aerial images. However, after a disaster such as a tornado or an earthquake many features in aerial images like monuments and unique buildings may change, and the image-based localization would become hard or even impossible. Consequently it is important to find features that remain unchanged or with fairly small changes, and can be detected both before and after a disaster. We have used a recent method for street detection from aerial images and shown that road networks and segments are disaster invariant and could be utilized for localization and mapping. The algorithm has been implemented and tested on satellite images from Google, with nearly equivalent resolution to aerial images. The successful result of detecting this DIF on Port-au-Prince, in Haiti, images before and after the recent earthquake is presented.**

I. Introduction

Unmanned Aerial Vehicles (UAVs) have various applications in tasks such as search and rescue, telecommunication, surveillance, environment and traffic monitoring systems. The fact that they are unmanned enables them to work in environments that are dangerous and/or inaccessible to humans. However this advantage comes with the extra cost of being autonomous. For instance, they should be able to localize themselves in an environment using landmarks and features. During recent decade, *Global Positioning System* (GPS), *Inertial Navigation System* (INS), and rangefinders have been the most popular tools for Ground Vehicle localization and pose estimation. But, they all have their own drawbacks. GPS, which includes measurements of the robot pose, is vulnerable to interference and jamming and even intermittently losing connection (without any obvious reason) for long periods of time [10]. INS, which is a set of accelerometers and gyroscopes and is used for motion estimation, is subject to well-known drifting errors due to integrations. Finally, it is hard, even impossible, to use laser on UAVs because of the vehicles' high altitude and the high amount of required power for operation of laser rangefinders.

Beside all the above shortcomings for these sensors, it is always possible that a sensor fails and having an alternative sensor for localization would be desirable. That is why vision has been selected in this work as a complement or a replacement for the other sensors, especially GPS, for UAV localization. Also, aerial images are very useful, especially because the chance that the view is occluded by an unknown object is very low, compared to images taken by Ground Vehicles, and the features in aerial images can be detected robustly from time to time for localization and SLAM.

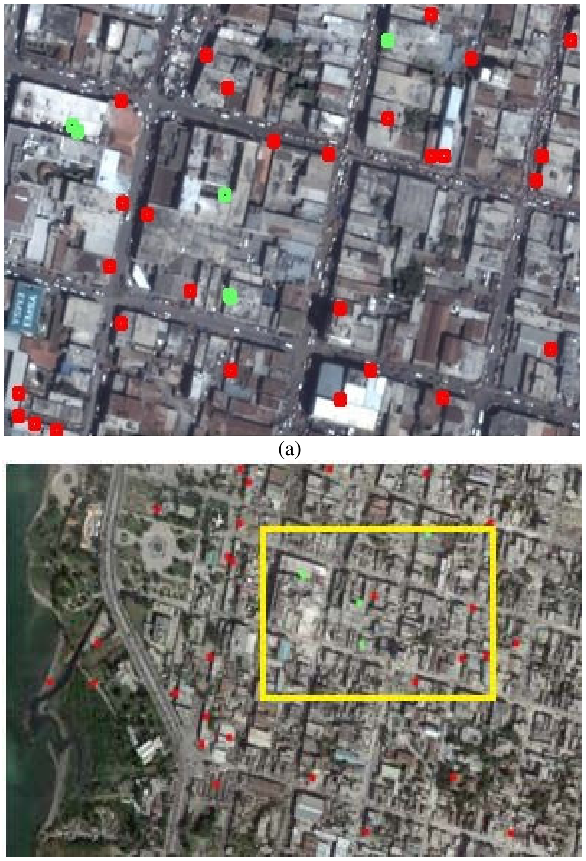
Visual features can be utilized in UAV localization and navigation in two ways. In first approach, usually used for micro UAVs, landmarks are put on the vehicle and a stationary camera on the ground detects them. The ground system remotely processes the data and performs the UAV localization and the estimated location is sent back to the UAV [1], [2]. This approach has limited applications, because finding the landmark on a vehicle flying outdoors is very difficult. More importantly, it requires an established platform on the ground to operate, which makes it nearly impossible to be employed in disaster stricken areas. In the second approach, the vehicle is self-contained and the camera is located on the vehicle. The vehicle detects the landmarks and features in images [3] captured by the camera and use them for localization or mapping. Our research is based on this approach.

Various visual features have been utilized for the purpose of localization, such as: *corners* [4], [5], *Weighted Grid Orientation Histogram* [6], [7], *Weighted Grid Integral Invariant* [8], and *Scale-Invariant Feature Transform* (SIFT) [9]. However, almost none of these methods are expected to perform satisfactorily in disaster stricken areas. After a disaster such as an earthquake, the structural characteristics of the environment change drastically, and it is even possible that some of the existing landmarks and prominent features of the environment become non-existent. This issue has been tested on SIFT features and illustrated in Fig. 1.

An image (Fig. 1a) is taken from the city of Port-au-Prince, Haiti before the earthquake in Januray2010. Another larger image (Fig. 1b) is taken after the earthquake which includes the original region and its neighborhood. SIFT features of the two images are found independently and compared together. The result of this SIFT matching should give us the region in the second image which is most probable to coincide with the first image (the correct region is shown by a yellow rectangle). The ideal situation is that the SIFTs in the first image only be matched to SIFTs inside the yellow-boundary region, so that we can deduce the correspondence of the first image and the yellow-boundary region in second image. In practice, however, out of 35 matched pairs (shown with red and green rectangles) in

B. Soleimani, M.H.Zokaei Ashtiani, B. Haji Soleimani, and H. Moradi are with Control and Intelligent Processing Center, ECE Dept., University of Tehran, Iran. Corresponding author: b.soleimani@ece.ut.ac.ir,

978-1-4244-6676-4/10/$25.00 ©2010 IEEE



(a)

(b)

Fig. 1. The found matched pairs in two images of Port-au-Prince, Haiti. a) An image taken before the earthquake, and b) the image of the same region with its neighborhood, taken after the earthquake. The green marks show the correct matched SIFT feature pairs, whereas the red marks indicate incorrect matched pairs.

these two images, only 5 of them are correctly matched (green rectangles). More importantly, the incorrect matched pairs are found all over the second image. This shows that we cannot infer anything from the distribution of matched pairs, and it is almost impossible to match an image taken after disaster with an original image before the disaster, solely from the SIFT features. It should be noted that after comparing the SIFT features of Fig. 1a only with the SIFTs in the yellow-boundary region in Fig. 1b, nearly 20 matching was found. Thus, the matching performance is higher when the the true correspondence in known.

In this study, we propose a Disaster Invariant Feature (DIF), i.e. roads and streets, for UAV localizations. We argue that road networks are unique features that can be detected from the air and used for UAV localization and mapping. More importantly, these features are detectable even after a disaster, such as an earthquake or a tornado, hits an area. We have implemented an especial street detection method to make use of such a DIF in a Bayesian filter based framework to perform the localization of a simulated blimp. It is important to mention that we intentionally did not consider other sensors such as GPS to highlight the effectiveness of this approach in the absence of the other sensors.

II. RELATED WORKS

As mentioned earlier, GPS, INS and range finders have been used for localization. There have been some efforts in order to combine these methods and benefit from all of them [11], [12], but the possibility of not receiving correct GPS signals does not vanish. Other sensors, such as altimeters, pressure sensors, sonars, and laser rangefinders [13], have also been used for pose estimation, mostly for altitude estimation. As mentioned earlier sonars and rangefinders have restricted applications due to high operation altitude of the UAVs and high required power of these sensors.

In recent years, vision sensors, both mono and stereo, have become more popular. Kendoul *et al.* [14] combined optical flow of a single camera with *Inertial Measurement Unit*. In [4] and [5] camera is used in combination with INS. They both have used *corner* detectors, and tried to find the current pose based on previous pose by matching the points in two images. However, the correspondence is still a problem, as well as the problem of drifting that may happen as a result of error in motion estimation. As a result, the use of these methods in disaster stricken areas is very limited.

It is possible to perform localization solely using cameras [15]-[17]. Besides the fact that images of a camera are visual sensory data, it should be noted that vision can also be used for motion estimation and state prediction. In fact, most of the localization methods that use vision are based on visual odometry [18]; assuming the initial pose is known, for instance from GPS, it is possible to find the pose in each step by matching features in the current and previous images. Again, two major problems arise from this; the correspondence of the feature points, and the accumulated positioning error (drift error) due to incorrect matching or external disturbances existing even in a single step. In [15] a solution is proposed that requires at least four features to be recognizable in each image. In [17] a similar homographic approach was implemented that used *corners* as features. Caballero *et al.* [16] showed that drift error could be reduced by using online mosaicking (instead of point matching)

In this work we have used vision to complement or replace sensors such as GPS in case of error in sensing or failure in the sensor. We propose a Disaster Invariant Feature which does not suffer from disasters such as earthquakes, a problem that exists with other features like SIFT or corners.

III. STREETS OR ROADS: A DISASTER INVARIANT FEATURE (DIF) FOR AERIAL VEHICLE IMAGES

As mentioned before, it is important to find features that remain nearly unaffected and are detectable both before and after a disaster. In our experiments, streets or roads have shown significant resistance to these types of disasters. Furthermore, the streets or roads have adequately wide distribution in the aerial images. Thus, even after destruction of some road segments, the structure of the road network can still be detected. In the following sub-sections we will show how the roads are detected and how successfully they are used for localization of a simulated autonomous blimp.

A. Road Detection

There exist few methods for detection of streets in aerial images, such as the ones described in [20]-[22]. Most of these methods are semi-automatic and need human assistance to perform satisfactorily. Many of them take too much time to be applicable in real-time applications, and some of them are only tested on particular images (in which the streets are easily detectable by using trivial methods, such as segmentation in HSV space). However, we are interested in automatic methods that are able to extract roads even in complex urban areas. In this work, we have exploited the method proposed in [23] which is described here briefly. The idea is to gradually expand a tree that tracks roads and identifies road networks. The root of the tree is planted on the image and it tries to grow and follow the directions that are more likely to contain roads based on two characteristics of them: 1) low variance in brightness along the roads and 2) the limited changes of direction while tracking the roads. In other words, this method utilizes both local features (variation in brightness) and global features (geometry of the whole road structure) to construct appropriate tree. A local search is used to track the roads in different possible directions while global constraints guarantee the smoothness of the roads.

The algorithm starts with selecting a seed, i.e. a single node or pixel, randomly on the image. In each iteration, an unexpanded node on the tree, with the highest probability of being on a road, is selected. Starting from that node, a few new hypothetical edges, i.e. expanded branches, are created, each of them corresponding to a new hypothetical node, and are given scores which reflect the likelihood that each of the edges are laid on a road. From the whole set of hypothetical nodes, the node with the highest score would be the winner and is selected for further expansion. This process continues until some termination criteria, such as the number of nodes in the tree, the ratio of the explored regions of the image to the whole image, the estimation of the area of the tree, or the minimum value for score of a new node, are satisfied. In our implementation, the number of nodes termination criterion is used to ensure the real-time performance of the algorithm. The maximum number of nodes in the tree is set experimentally (and roughly) by running the method on some sample outdoor images. The score of a node, say the i^{th} node is computed by the following formula:

$$score(i) = \alpha.\frac{\cos\left(\left|\theta_i - \tilde{\theta}_{p(i)}\right|\right)^{\beta}}{B(p(i), i)} + (1-\alpha).score(p(i)) \tag{1}$$

where $p(i)$ is the parent of the i^{th} node and θ_i is the direction of the edge connecting node i and its parent. The cosine term reflects deviation from the weighted average direction of its parents ($\tilde{\theta}_i$), and is computed by:

$$\tilde{\theta}_i = \gamma.\theta_i + (1-\gamma).\tilde{\theta}_{p(i)} \tag{2}$$

in which γ is a discount factor between 0 and 1 and determines the sensitivity of the mean direction of the tree to direction of the last branch. We heuristically set it to 0.5 to let the tree change direction when it reaches a turning point. The brightness score is computed based on the following formula

$$B(i,j) = \sum_{k \in s_{ij}} |b_k - b_{k-1}| \tag{3}$$

in which s_{ij} determines the sequence of pixels on the line segment between node i and j, and b_k points out the brightness of k^{th} node on this sequence. As it can be seen, $B(i,j)$ computes the sum of differences of brightness's of consecutive points along an edge which is a measure of variance in the brightness of the pixels on it. The trade-off between brightness score and direction score is provided by β, whose value can be set either experimentally or heuristically, so that these two scores take their own desired parts in the overall score. Also, α, which is between 0 and 1, is a balancing factor that determines the significance of the parent node's score, relative to calculated brightness and direction scores, and has been set to 0.5 in our experiments. However, setting β requires more considerations. For this purpose, the direction score and the brightness score may be calculated experimentally for trees in some target images, and normalized together. Based on this, we concluded that β should be more than 0.7, and we set it to 0.8. It is worth mentioning that as we will discuss in next subsection, the detection needs not to be perfect for utilization in the correction phase of a Particle Filter for localization.

B. Constructing Feature Maps

Based on detected roads, a *feature map* can be constructed. The idea is to compare this map with the *global map* of the environment, and use the result in a Bayesian filter based framework. Generally, maps are described in two ways: *vector maps* and *raster maps*. In first group, map is represented with the use of some geometrical primitives, like points, lines, etc., whereas maps in the second group, similar to occupancy grid maps, are consisted of cells or tiles with modifiable resolution. Global map of the environment and feature map could be represented in both ways; however, we use grid-based maps, to represent both global and feature maps, because they are less sensitive to local changes in the streets, and also they could be compared with each other more easily and more efficiently.

Global maps can be constructed in two ways; they can be derived from existing CAD maps or drawn manually, or they can be made by applying the same street detection method on old images. Since the global maps are available in advance, the improved version of road detection methods can be applied w/o worrying about the real-time constraints. This offline nature of processing the global maps allows handling the huge size of these maps realistically. The certainty of having a street in a cell in the resulting grid-based maps, both the global maps and the feature maps, varies between 0 and 1.

As stated before, the result of matching the feature map with the global map may be used in the correction phase of a Bayesian filter. The justification is as follows: the position and orientation of the camera in the environment corresponds to a pixel in the global map (barring its altitude). The camera is fixed on the UAV, and its location is related to the UAV pose that we want to find. This pose is not known precisely, but only an estimation of it is in hand. This is where Bayesian filter framework would be advantageous. In each step, the estimation of the UAV pose is used to determine the region on the global map by which the local map is compared. Then a similarity measure is calculated and is used to modify the estimation of the UAV pose. This prediction-correction loop is done recursively during normal operation of the UAV.

IV. Simulation and Discussion

A. Particle Filter

Bayes filter is an algorithm for calculating beliefs, i.e. posterior probabilities of state variables computed after incorporating the available data. Usually these state variables are robot's pose and/or speed, and the measurements taken by robot's sensor are considered as data. Each iteration comprises two steps; in *prediction* step, prior estimation of the state is changed based on the current input (*motion model*), whereas in *correction* step, this estimation is updated using sensory data (*measurement model*). Particle Filter [24] is a non-parametric implementation of Bayes Filter. Briefly speaking, it consists of a set of particles, each of them representing a hypothesis of the state. In each step, particles are relocated based on motion model, i.e. $p(x_t|u_t, x_{t-1})$, and weights are assigned to them according to measurement model, i.e. $p(z_t|x_t)$. Here u_t, x_t, and z_t denote input, state, and measurement data at time t. After that, particles are *resampled* based on their weights, and as a result, their distribution changes and becomes more similar to belief probability distribution. To implement Particle Filter, motion model and measurement model should be defined. In the remainder of this section, these models are presented.

Motion Model: in our study, we used a UAV with three motors: two of them for planar motion (right and left motors) similar to a planar mobile robot, and the third for changing the altitude of the UAV. We can write Newton-Euler equation of motion for the UAV, and derive its motion model (for example, this has been done for a blimp by Zufferey *et al.* [25]). In our current implementation we ignored the third motor and assumed the altitude of the UAV does not change over time. It is also usual to assume that the motion is parallel to the ground. Therefore, the motion model becomes very simple, and in fact it could be written similar to that of a planar wheeled mobile robot (WMR):

$$\begin{aligned} \dot{x} &= v\,.\cos\psi \\ \dot{y} &= v\,.\sin\psi \\ \dot{\psi} &= w \end{aligned} \tag{4}$$

Assuming independent control for right and left motors, it is possible that $v = 0$ and $w \neq 0$, which results in vehicle rotating in its position. Furthermore, uncertainty can be modeled by assuming that v and w are random variables (usually Gaussian) instead of deterministic inputs. It should also be noted that the use of other motion models is also possible and does not have any major effect on the measurement model, which is the main contribution of this paper.

Measurement Model: in our method, each measurement is represented by a binary matrix, m_{local}, which is computed from detected roads. In our road detection implementation, it is computed by counting the number of tree nodes in each tile of the feature map, and setting the tiles with number of nodes above a threshold to 1 (*on*) and the others to 0 (*off*). This local matrix is then compared to the corresponding section of the global map (m), and a similarity measure is computed. We use the map matching formula introduced in [24]:

$$\rho_{m,\, m_{local}, x_t} = \frac{\sum_{x,y}\left[\left(m_{x,y} - \bar{m}\right).\left(m_{x,y,local}(x_t) - \bar{m}\right)\right]}{\sqrt{\sum_{x,y}\left(m_{x,y} - \bar{m}\right)^2 \sum_{x,y}\left(m_{x,y,local}(x_t) - \bar{m}\right)^2}} \tag{5}$$

$$\bar{m} = \frac{1}{2N}\sum_{x,y}\left(m_{x,y} + m_{x,y,local}\right) \tag{6}$$

$$P(m_{local}|x_t, m) = \max\left(\rho_{m,\, m_{local}, x_t}, 0\right) \tag{7}$$

where x and y subscripts determine the position of the pixel in the image, and $P(m_{local}|x_t, m)$ is exactly the distribution that new samples are derived from. Both m and m_{local} in the above equation accept values between 0 and 1 (not just 0 or 1); in other words, a detection method with probabilistic outputs can also use these weighting formulae.

B. Results

Fig. 2 shows the results of the road detection algorithm, implemented on two images taken before and after earthquake from Port-au-Prince, Haiti. The road detection algorithm is performed on the image, and the resulting occupancy grid map is constructed. Each tile is 10*10pixels. Fig. 3 shows a sample result of performing the localization on a test image taken from Las Vegas, Nevada. Fig. 4 depicts the same results, but the images are zoomed-in around the true pose. In both images, particles are shown by red dots. Different steps of the localization are shown. Final true pose is the pixel (649,920) and it is estimated as (659,922) by the Particle Filter, which shows only a slight error in localization (about 10m, considering the level of the images we have used), as can be seen from the position of green circle (the blue ellipse is hidden behind the mass of red dots). It should be noted the original image's resolution is 1200*1200pixels. Furthermore, in each step of the PF, instead of resampling all particles, %15 of the particles are regenerated uniformly in the state space, to achieve better exploration of the state space and reduce the effect of particle deprivation problem [24]. Each step of the particle filter -consists of applying motion model to particles, street detection on an image, weighting the particles using map matching, and finally resampling the particles- takes about 2seconds to be completed on these images (on a 2.80GHz Pentium 4 CPU with 1MB Cache and 1GB RAM). The required time will be very lower for images with lower resolution. However, even in this state (without

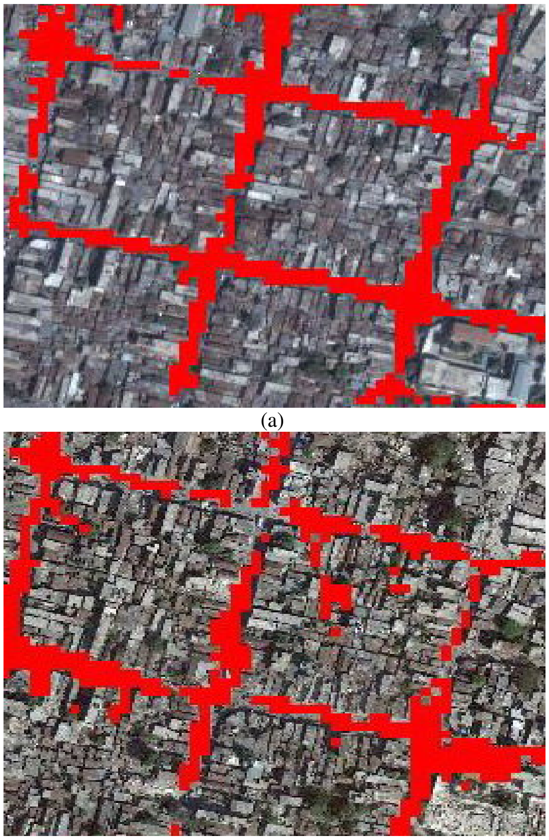


Fig. 2. The result of street detection method on an image taken a) before and b) after earthquake from Port-au-Prince, Haiti.

doing optimizations of the algorithm) it can be used for low-speed vehicles, such as blimps (but not on airplanes and helicopters). The bottleneck lies on street detection and map matching, which currently are performed in all steps of the PF. It should be noted that this is not necessary. In fact, we can implement the algorithm such that these are performed only once in many consecutive steps. This further reduces the time complexity of the algorithm.

V. Discussion and Future Works

In this paper we presented a novel Disaster Invariant Feature (DIF) that can be detected before and after a disaster such as an earth quake or a tornado. Such a feature can be used for the localization of aerial vehicles, especially for the purpose of search and rescue. It is important to mention that the existence of sensors such as GPS does not eliminate the need for localization using vision since any sensor such as GPS may not be accessible for a period of time or fail entirely. The proposed DIF has been successfully used to localize a simulated blimp on a real disaster stricken area.

Future works include improving the street detection method and further testing in various disaster stricken regions. Also the construction of probabilistic feature maps would be investigated. Furthermore, we will deploy the framework on a real blimp to test it in different natural circumstances.

VI. Acknowledgement

The authors want to thank Amir Hossein Bakhtiary for his assistance in the SIFT analysis.

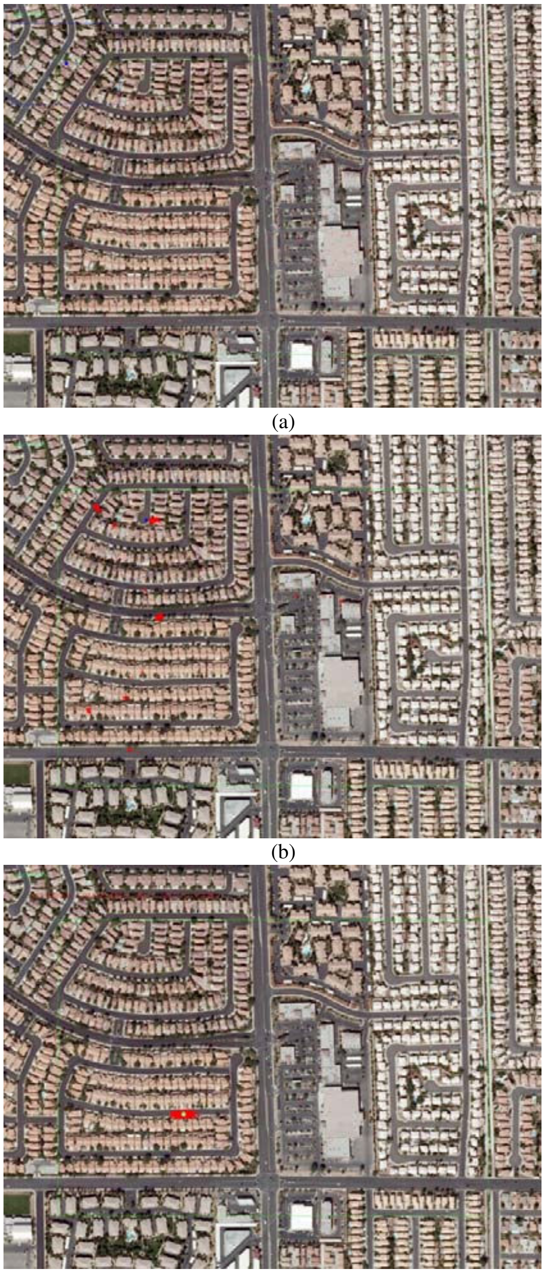
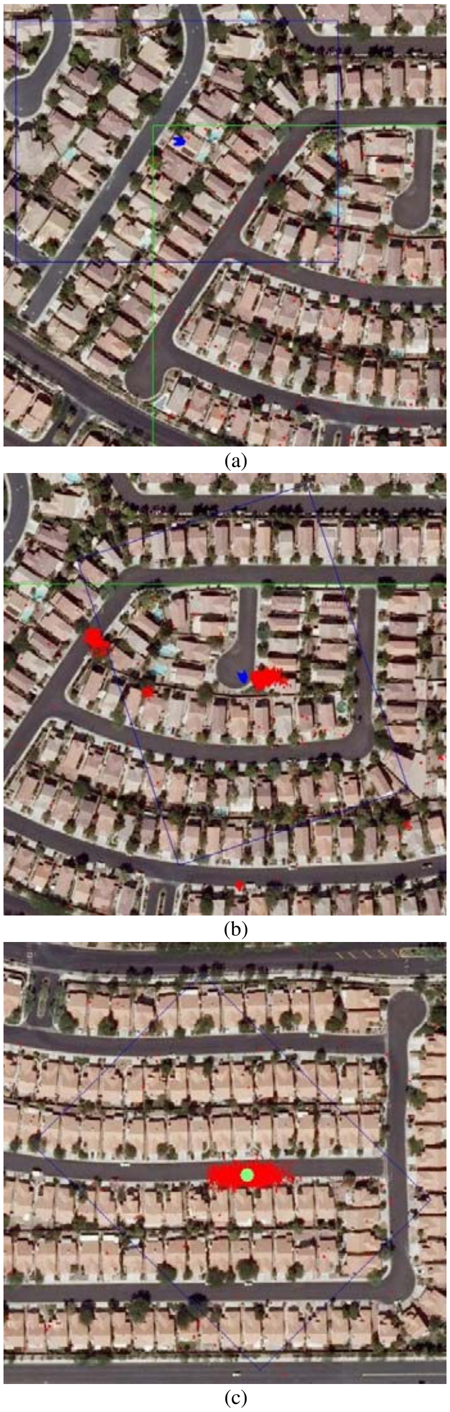


Fig. 3. The result of performing Monte Carlo Localization of a UAV on a sample image taken from Las Vegas, Nevada, during different steps. a) initial step: particles are uniformly distributed; b) a middle step (step 80): many different hypotheses coexist; c) last step (step 100) only two hypotheses are remained, and the best one is closer to the true pose. The red dots are the particles, the green arrow-like ellipse indicates the true pose, and the big green circle shows the estimated pose based on particles. The blue and green rectangles are window viewed by the camera in each step, and the acceptable region for particle generation and resampling in all steps.



(a)

(b)

(c)

Fig. 4. The result of performing Monte Carlo Localization of a UAV on a sample image taken from Las Vegas, Nevada, during different steps. a) initial step: particles are uniformly distributed; b) a middle step (step 80): many different hypotheses coexist; c) last step (step 100) only two hypotheses are remained, and the best one is closer to the true pose. The red dots are the particles, the green arrow-like ellipse indicates the true pose, and the big green circle shows the estimated pose based on particles. The blue and green rectangles are window viewed by the camera in each step, and the acceptable region for particle generation and resampling in all steps.

REFERENCES

[1] J. Kim, M.S. Kang, and S. Park, "Accurate Modeling and Robust Hovering Control for a Quad–rotor VTOL Aircraft," *Journal of Intelligent and Robotic Systems*, vol. 57, Jan. 2010, pp. 9–26.

[2] L.C. Mak, M. Whitty, and T. Furukawa, "A localisation system for an indoor rotary-wing MAV using blade mounted LEDs," *Sensor Review*, vol. 28, 2008, pp. 125–131.

[3] F. Andert, F.M. Adolf, L. Goormann, and J.S. Dittrich, "Autonomous Vision-Based Helicopter Flights Through Obstacle Gates," *Journal of Intelligent and Robotic Systems*, vol. 57, Jan. 2010, pp. 1–22.

[4] P. Corke, "An inertial and visual sensing system for a small autonomous helicopter," *Journal of Robotic Systems*, vol. 21, 2004, pp. 43–51.

[5] M. George and S. Sukkarieh, "Camera Aided Inertial Navigation in Poor GPS Environments," *2007 IEEE Aerospace Conference*, 2007, pp. 1–12.

[6] D.M. Bradley, R. Patel, N. Vandapel, and S.M. Thayer, "Real-time image-based topological localization in large outdoor environments," *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2005, pp. 3670–3677.

[7] S. Erhard, K.E. Wenzel, and A. Zell, "Flyphone: Visual Self-Localisation Using a Mobile Phone as Onboard Image Processor on a Quadrocopter," *Journal of Intelligent and Robotic Systems*, 2009, pp. 1–15.

[8] C. Weiss, A. Masselli, H. Tamimi, and A. Zell, "Fast outdoor robot localization using integral invariants," *Proc. 5th International Conference on Computer Vision Systems (ICVS)*.

[9] A. Irschara, C. Zach, J.M. Frahm, and H. Bischof, "From structure-from-motion point clouds to fast location recognition," *Proc. CVPR*, 2009.

[10] J. Volpe, "Vulnerability assessment of the transportation infrastructure relying on the global positioning system," 2001.

[11] J.Z. Sasiadek and P. Hartana, "Sensor fusion for navigation of an autonomous unmanned aerial vehicle," *IEEE International Conference on Robotics and Automation (ICRA). Proceedings*, 2004.

[12] N. Abdelkrim, N. Aouf, A. Tsourdos, and B. White, "Robust nonlinear filtering for INS/GPS UAV localization," *Control and Automation, 2008 16th Mediterranean Conference on*, 2008, pp. 695–702.

[13] A. Shaw and D. Barnes, "Landmark recognition for localisation and navigation of aerial vehicles," *IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS). Proceedings*, 2003.

[14] F. Kendoul, I. Fantoni, and K. Nonami, "Optic flow-based vision system for autonomous 3D localization and control of small aerial vehicles," *Robotics and Autonomous Systems*, vol. 57, 2009, pp. 591–602.

[15] K. Kaiser, N. Gans, and W. Dixon, "Localization and control of an aerial vehicle through chained, vision-based pose reconstruction," *American Control Conference (ACC)*, 2007, pp. 5934–5939.

[16] F. Caballero, L. Merino, J. Ferruz, and A. Ollero, "Unmanned Aerial Vehicle Localization Based on Monocular Vision and Online Mosaicking," *Journal of Intelligent and Robotic Systems*, vol. 55, 2009, pp. 323–343.

[17] F.M. Da Silva Metelo and L.R.G. Campos, *Vision Based Control of an Autonomous Blimp (VIDEOBLIMP)*, Universidade Tecnica de Lisboa, Instituto Superior Tecnico, 2003.

[18] O. Amidi, T. Kanade, and K. Fujita, "A visual odometer for autonomous helicopter flight," *Intelligent autonomous systems: IAS-5*, 1998, p. 123.

[19] I. Ulrich and I. Nourbakhsh, "Appearance-based place recognition for topological localization," *IEEE International Conference on Robotics and Automation*, 2000, pp. 1023–1029.

[20] J.B. Mena, "State of the art on automatic road extraction for GIS update: a novel classification," *Pattern Recognition Letters*, vol. 24, 2003, pp. 3037–3058.

[21] J.B. Mena and J.A. Malpica, "An automatic method for road extraction in rural and semi-urban areas starting from high resolution satellite imagery," *Pattern recognition letters*, vol. 26, 2005, pp. 1201–1220.

[22] J. Hu, A. Razdan, J.C. Femiani, M. Cui, and P. Wonka, "Road network extraction and intersection detection from aerial images by tracking road footprints," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, 2007, pp. 4144–4157.

[23] M.H. Zokaei Ashtiani, *Street Detection in Aerial Images Using Expanding Trees*, University Of Tehran, 2010.

[24] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics. 2005*, MIT Press, .

[25] J.C. Zufferey, A. Guanella, A. Beyeler, and D. Floreano, "Flying over the reality gap: From simulated to real indoor airships," *Autonomous Robots*, vol. 21, 2006, pp. 243–254.