Sensorcam cameras and image registration

Zichong Chen
LCAV, EPFL, Switzerland

I. Sensorcam Prototype v.1

Our first approach to visual monitoring was the deployment of mobile phone an integrated camera. It was powered by a car battery and installed on the unknown glacier since unknown date. It kept sending one picture of the glacier every hour during daytime and stopped working after two months due to power depletion. We consider it as a successful feasibility study.

Based on the experiences gathered in this first deployment, we developed *Sensorcam*, a more flexible camera designed to be fully autonomous. It includes a Colobri PXA270 embedded platform, a GSM radio communication module, and a solar power system as the energy source. The master board runs an embedded Linux system 2.6.26 and controls all the peripherals.

Fig. 1a shows the Colobri PXA270 board, the GSM module, and the associated "glue" board. To provide power supply, we build a solar power system with a battery and a 2.4W solar panel (Fig. 1b). We deployed this solar system at an Alps mountain for more than six months. Using solar radiation data collected from the deployed system, we calculate the average energy supply during the winter period (most adverse period) from 15th Jan 2011 till 17th Mar 2011. The energy budget is listed in Table I.

II. Image Registration

To establish point-wise correspondence between stereo-view images, epipolar geometry is the standard tool for image registration [5]. However, it requires known scene structure in prior, which is usually not feasible. Estimating depth is one way to retrieve scene structure. However, the state-of-the-art depth estimation algorithms are not accurate enough for registration task. As an alternative, we propose a registration algorithm based on the homography geometry. This algorithm is used in our works [2] [3] for image registration.

The homography geometry is described by a single 3×3 matrix, so that any correspondence points $\mathbf{x}, \mathbf{y}$ of stereo-view images can be related by a linear transformation[1]:

$$\mathbf{x} = \mathbf{H} \cdot \mathbf{y}.$$

The simplicity of this model greatly reduces the complexity of model estimation and the transmission overhead. On the other hand, such a model can be inaccurate because it is designed for (i) two images taken by a fixed rotating camera or (ii) stereo-view images with only planar scenes, which is not the general case. The registration error in the near field scenery creates a negative effect for the overall coding efficiency. To overcome this, we propose a robust algorithm to detect a common imaging area that approximately obeys the homography geometry. This strategy fits our application, since the planar scenes "far" from the cameras dominates in environmental monitoring applications. We focus the distributed successive refinement on the common imaging area, the rest of the image have very limited correlation and therefore can be coded independently.

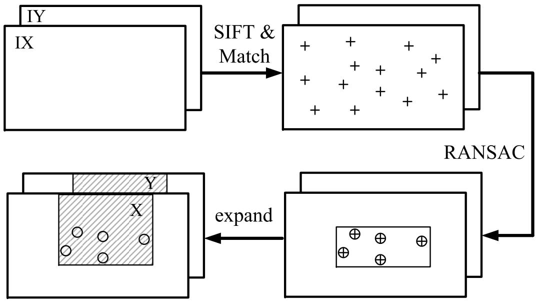


Fig. 2. Robust registration algorithm of stereo-view images: the last two steps determine the common imaging area. Cross denotes the original point tagged by SIFT/SURF and circle denotes the inliers selected by RANSAC.

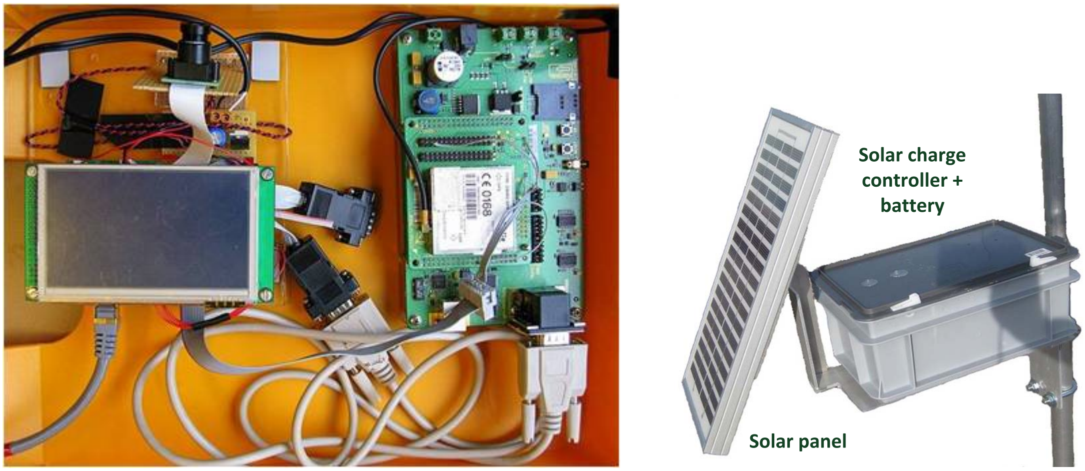
To estimate the homography model (matrix $\mathbf{H}$) for two stereo-view images, we need at least 4 non-collinear correspondence point pairs. These correspondence pairs are obtained by extracting feature points using SIFT [6] or SURF [1] and then establishing the correspondence using the nearest neighbor search method. To avoid outliers in the *correspondence poll* we generated, we use RANSAC [4] as a robust model estimation algorithm to overcome the errors in the dataset and keep time efficiency at the same time.

To automatically detect the common imaging area, we can use the fact that the feature points selected by RANSAC tends to cluster in practice. These points will define the area that is well registered after homography transformation. Still, this simple idea may miss some background area (e.g., sky) if the features of the texture are too weak to be selected. However, in environmental monitoring applications, the background usually lays in the upper part of image. Therefore, we can naturally assume the region above the labeled area is the background (e.g., the sky), which is also a planar scene.

In sum, Fig. 2 illustrates the entire routine of robust registration algorithm (Fig. 3 shows sample registration results):

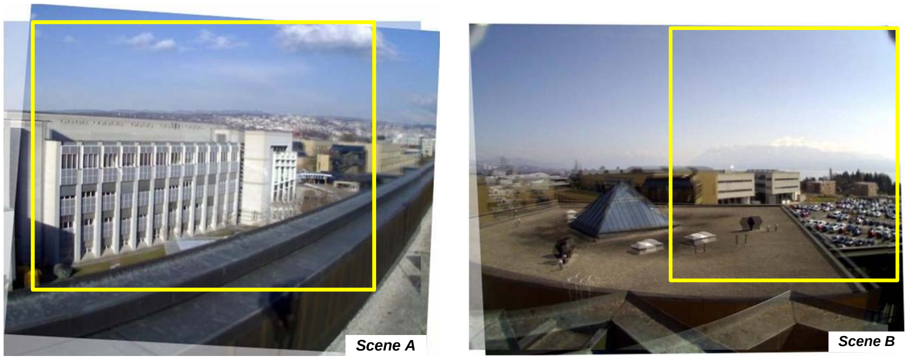
1) Feature points are generated for two stereo-views image IX and IY, and matched with each other.
2) RANSAC selects the inlier point pairs that gives a robust estimation of the homography matrix.

[1] $\mathbf{x}, \mathbf{y}$ are homogeneous coordinates, see definition at [5].



(a) (b)

Fig. 1. *Sensorcam* prototype v.1: (a) Master board and GSM module. (b) Solar power system.



(a) (b)

Fig. 3. Correspondence mapping using homography geometry of stereo-view images from datasets Scene A and Scene B. The registered right view is overlaid on left view. The yellow frame labels the area that is well registered.

3) The minimal rectangular hull containing all inlier point pairs is determined, and then extended to the top of the image. This is the detected common imaging area X, Y.

III. Experimental Setup of [3]

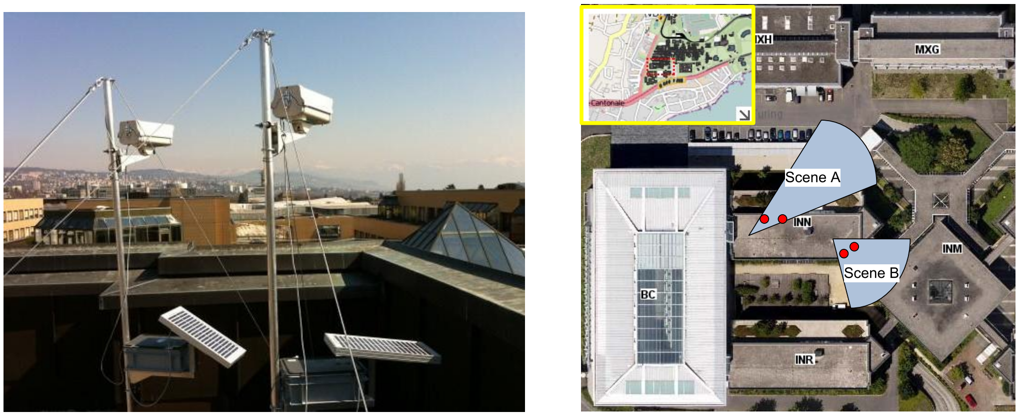
To analyze the performance for the proposed algorithm in [3], we first collect image data-sets using conventional cameras, and then run the algorithm on the *Sensorcam* v.1 board to get a approximate energy profiling. Table I lists some of the parameters related to the experiments. In this section, we introduce the experimental setup, and also give the justifications how the simulation represents a practical system.

A. Interleaved sampling

Fig. 4a shows the two cameras deployed on the roof of a building with the solar power system. Each camera can be programmed to capture images periodically and store them in a local SD card. The two deployed cameras are synchronized to start simultaneously and sample at one-minute interval. By selecting corresponding images, we can get the datasets as in the interleaved sampling setup. Moreover, by subsampling the video stream, we get datasets with different sampling frequency f.

TABLE I
PARAMETERS USED IN ENERGY PROFILING

`Colibri PXA270` [8]	
CPU	520MHz
SDRAM	64M
FLASH	32M
Consumption (normal)	800mW
Consumption (sleep)	7mW
`long range radio` [9]	
Consumption (Tx@30dBm)	2860mW
Consumption (Rx)	240mW
Consumption (idle)	14mW
Speed	16Kbps
`camera module OV7720` [7]	
Resolution	640×480
Consumption (active)	120mW
`solar power system @ 2.4W`	
Battery capacity	135KJ
Battery charge efficiency	70%
Average energy supply (winter)	12.6KJ/day



(a)

(b)

Fig. 4. Experimental setup to capture datasets: (a) Two surveillance cameras with solar power system watching Scene B. (b) The setup of two cameras (red dots) on roof of a building in campus. Scene A and B show the viewpoints of two datasets.

B. Datasets

In order to investigate the algorithm in a more comprehensive way, we deploy the cameras to monitor two different sceneries (see Fig. 4b) with opposite depth structures: Scene A captures the facade of a building which is a planar scene suitable for homography geometry. In contrast, Scene B includes buildings stretching out to the mountains far away, which has complex depth structure.

TABLE II
PARAMETERS OF DATASET COLLECTING

dataset	sample	weather	duration	frames
Scene A	Fig. 3a	cloudy	14:00-24:00	594
Scene B	Fig. 3b	sunny	06:30-16:30	600

The related information of these two datasets are listed in Table II, and all raw images can be accessed at http://dl.dropbox.com/u/7084673/papers/multicamera_datasets.zip.

C. Overhearing

We calculate if there is enough "idle" time in channel usage for overhearing. From Table I, the average energy budget we have per day after canceling out background consumption during idle/sleep mode is $12.6\text{KJ} \times 0.7 - (7+14)\text{mW} \times 24\text{h} = 7.0$ KJ. For a video with resolution 640×480, assuming we use the naive Motion JPEG to compress the video, the typical size of each frame ranges from 20 to 40 KB. The consumption per frame can be estimated as:

- TX: $40\text{KB}/16\text{Kbps} \times 2860\text{mW} = 28.6 \sim 57.2\text{J}$.
- Camera: $1\text{s} \times 120\text{mW} = 0.12\text{J}$.
- CPU (capture/compress image): $2\text{s} \times 800\text{mW} = 1.6\text{J}$.
- Overhead (GSM startup, etc.): $10\text{s} \times 1\text{W} = 10\text{J}$.

Therefore, the energy constraint allows a sampling frequency (sleep at nighttime) at $8 \sim 14$ pic/h. In other words, the sampling interval of each camera is $250 \sim 450\text{s}$. To transmit an image, it takes $20 \sim 40\text{KB}/16\text{Kbps} = 10 \sim 20\text{s}$. Thus, it leaves enough idle time: $20\text{s} \ll 250\text{s}$.

D. Algorithm implementation

We implement all algorithms described (except DISCOVER) using C/C++ and run them on the *Sensorcam* v.1 board. They are based on several open-source libraries, including:

1) OpenCV: library of programming functions for real time computer vision.[2]
2) x264: library and application for encoding video streams into the H.264/MPEG-4 AVC format.[3]

The executable codec of DISCOVER is also available on http://www.discoverdvc.org/, but not applicable for running on the *Sensorcam* v.1 board due to lack of source code.

E. Energy profiling

The most critical factor in evaluating a wireless camera network is the energy consumption. We can retrieve an approximate energy profiling of each camera by combining computation and communication consumptions:

1) The algorithms are executed on a *Sensorcam* v.1 board based on the datasets collected beforehand. Therefore the time spent on running the program can be known accurately, which indicates the energy used for computation.
2) Once we know the size of codewords generated by the algorithm, we can calculate the transmission time from GSM Tx speed, which indicates the energy used for communication (including passive overhearing).

As we point out in Sec. III-D, DISCOVER is an exception that cannot be executed on the *Sensorcam* v.1. To estimate its computation consumption, we can refer to the relative complexity of DISCOVER to H.264[4], and use H.264 consumption to determine the computation energy of DISCOVER on the *Sensorcam* v.1 board.

[2] See http://opencv.willowgarage.com/wiki/
[3] See http://www.videolan.org/developers/x264.html
[4] See http://www.img.lx.it.pt/∼discover/complexity.html

It is also worth mentioning that we only consider the energy consumption of wireless cameras, and the BS is assumed to have sufficient power supply. As a consequence, our system design principle is to minimize the overall energy consumption of the cameras, ignoring any extra energy consumption at the BS.

REFERENCES

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

[2] Z. Chen, G. Barrenetxea, and M. Vetterli. Distributed successive refinement of multi-view images using broadcast advantage. *submitted to IEEE Trans. Image Process.*, 2011.

[3] Z. Chen, G. Barrenetxea, and M. Vetterli. Share Risk and Energy: Sampling and Communication Strategies for Multi-Camera Wireless Monitoring Networks. In *Proceedings of the 31st Annual IEEE International Conference on Computer Communications (INFOCOM 2012)*, 2012.

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

[5] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge Univ Pr, 2003.

[6] D. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150 – 7, Los Alamitos, CA, USA, 1999.

[7] OmniVision Technologies, Inc. *OV7720 CMOS VGA CAMERACHIP SENSOR DATASHEET*.

[8] Toradex AG. *Colibri XScale PXA270 Datasheet* .

[9] A. Wang and C. Sodini. On the energy efficiency of wireless transceivers. In *Proceedings of IEEE International Conference on Communications*, volume 8, pages 3783–3788, 2006.