Markov Chain Monte Carlo Cascade for Camera Network Calibration based on Unconstrained Pedestrian Tracklets

Louis Lettry[1] Ralf Dragon[1] Luc Van Gool[1,2]

[1] CVL, ETH Zurich, Switzerland, [2] VISICS, KU Leuven, Belgium

Abstract. The presented work aims at tackling the problem of externally calibrating a network of cameras by observing a dynamic scene composed of pedestrians. It relies on the single assumption that human beings walk aligned with the gravity vector. Usual techniques to solve this problem involve using more assumptions such as a planar ground or assumptions about pedestrians' motion. In this work, we drop all these assumptions and design a probabilistic layered algorithm that deals with noisy outlier-dominated hypotheses to recover the actual structure of the network. We demonstrate our process on two known public datasets and exhibit results to underline the effectiveness of our simple but adaptable approach to this general problem.

1 Introduction

External calibration of a camera network is a process preceding many other tasks in computer vision such as scene reconstruction or human gesture analysis. Manual techniques are inconvenient as they require moving a calibration pattern in the common field of view or manually annotating corresponding points in multiple images. Automatic techniques exist to circumvent this manual part by automatically detecting corresponding keypoints or regions. However, when the baseline grows or the scene becomes dynamic, precision and recall of all kinds of such correspondences drop dramatically.

The use of dynamic objects finds its use in many different fields of work. Surveillance applications usually cannot rely on background correspondences of the scene as people are moving in front, or because of the lack of stable detectable keypoints. A variety of work has already been done in this area, relying on assumptions about different aspects of human beings such as average height, motion or a planar ground.

In this work, we aim at demonstrating the possibility to solve this problem by using the smallest assumption of human walking: Human beings stand against a common gravity vector while walking. Following the paradigm that *intra-camera* correspondences (between multiple time steps) are more reliable than *inter-camera* ones (e.g. pedestrian correspondences), we fit a plane through an estimated gravity vector of a tracked person at two time instances (Fig. 1). Using a probabilistic sampling framework, we establish plane correspondences between multiple camera pairs at different time steps and intervals in order to estimate the relative pose between two cameras. On a coarser level, all pairwise pose estimations will be fused into a geometrically consistent network.

We believe the contributions of our work to be multifold:

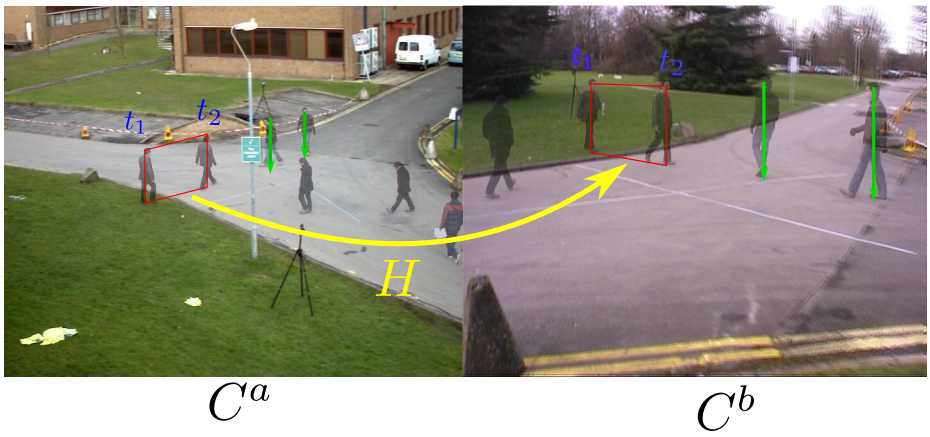

Fig. 1: Views from two cameras C^a and C^b at times t_1 and t_2 (stacked with transparency). In each camera view, a walking person defines a plane (marked in red) between his head and foot points in two time steps. Planes between multiple cameras are related by a homography H, which is used during our external camera calibration approach. Green arrows show gravity vector examples.

- A layered Markov chain Monte Carlo algorithm for camera calibration to work with noisy outlier-dominated data.
- Usage of foot-head planes for external camera calibration.
- Extension to the Shortest Triangle Paths Algorithm to incorporate stability constraints.

We present the current state of the art in this field in the section 2. Section 3 presents the underlying mathematical model used for relative pose estimation based on homography extraction and decomposition over four points on a plane. It is followed by the in-depth presentation of our algorithm in section 4. Afterwards will come the results presentation in section 5 before concluding in section 6.

2 Related Work

Camera calibration is extensively studied as knowledge about the scene geometry simplifies many computer vision tasks, e.g. multi-view object tracking or searching correspondences along the epipolar line for depth estimation. If we assume we have correspondences between image points $\bar{x_i^c}$ in different cameras C^j, we can use the regular structure-and-motion approach: combine all $\bar{x_i^c}$ in a measurement matrix which is factorized into the underlying 3D points $\boldsymbol{X}_i$ and the motion of the cameras.

Manual techniques have been created such as [1] which uses this technique with one light source manually moved through the scene, providing one inter-camera correspondence for many frames. In order to provide many correspondences per frame, calibration patterns could be used such as [2] for intrinsic parameters estimation or [3] in the case of a camera network. However, such manual procedures can be cumbersome or even impossible to conduct in certain situations, e.g. surveillance setups where

cameras are not reachable. An automatic and reliable procedure is thus needed for such situations.

Automatic inter-camera correspondences have been the solution developed to overcome those manual procedures. These techniques rely on detecting keypoints and establishing correspondences based on appearance of a patch around the extract points [4]. Non-linear optimizations can then be used to improve the precision and quality of such estimations [5]. However, determining inter-camera correspondences automatically becomes especially hard if cameras have a small field of view overlap (or even none in a network of cameras), or if they watch in different directions (also called *wide baseline*) as analyzed for example by [6]. To focus on the calibration approach, many methods take given correspondences as input [7, 8]. Very few methods do not require such knowledge, as [9].

Even though dynamic objects add another layer of complexity, they have been used and analyzed to extract information of all kinds. Pedestrians, for example, have been studied in many different fields of computer vision such as in detection task [10] or for tracking purposes [11]. They also have been extensively used as observations of inter-camera correspondences in camera calibration [12–19, 7, 20, 9, 8]. Although being intrinsically hard elements to work with due to their intra-class variety and per-instance non rigid deformations, they can be used in conjunction with various assumptions or priors.

Many different methods relying on pedestrian observations have been presented to solve the intrinsic parameters estimation problem. [17] suggested to extract vanishing points and line on a single pedestrian. Since vanishing points are sensitive to noise, other works have built upon it to robustify this approach, such as [18] which detects leg crossing events in order to extract more accurate foot-head points, or [14] which proposes a probabilistic approach in the form of a Markov-chain Monte-Carlo process to handle noise. [13] worked on a different assumption which incorporates a prior about the pedestrian motion. An important shadow model is added by [12] to extract more accurate points, with the same goal [15] used a human model.

External camera calibration has also been addressed using pedestrians as basic observations. [21] proposed to observe soccer players with PTZ cameras for calibration helped by the particular ground markings of soccer fields. [20] also calibrated PTZ camera networks but by tracking a single pedestrian walking on a plane, requiring a known inter-camera correspondence. On another side, [9] showed a work not requiring inter-camera correspondences by working directly on foreground blobs instead of pedestrians. All these works assumed a planar ground where people walked, [7] went away from this assumption and proposed camera network calibration on uneven terrains.

The underlying optimization process in camera calibration can take very different shapes to overcome noisy data. For example as a fully probabilistic approach as in [14]. Certain methods filter outliers based on robust analysis [7]. Others may want to extract only very reliable points by adding different models to compensate for noise as [12] which learns a complex shadow model to increase the precision of point extraction or [15].

In this work, we do not focus on precise point extraction or assume reliable and accurate information. This is motivated by the will of being robust in every situation, even when early processings in the pipeline, such as background subtraction or tracking, may fail. By extension, we do not use complexe non-linear refinment as it ask for accurate data in the first place.

The closest works to our approach are [14] and [9]. [14] proposed a Markov chain Monte Carlo approach but focused only on intrinsic camera calibration of one camera. We present a 3-layer cascade of Markov chain Monte Carlo in the different setup of external network calibration. [9] also follows the idea of analyzing inter-camera correspondences prior to network calibration but relies on a stable ground plane estimation from person heights.

3 Pose Estimation using Tracklet Correspondences

Our pose estimation is based on the decomposition of homographies $\mathbf{H_{ab}}$ into

$$\mathbf{H_{ab}} = \mathbf{K_b}(\mathbf{R_{ab}} + \frac{1}{d}\boldsymbol{t}_{ab}\boldsymbol{n}^\intercal)\mathbf{K_a^{-1}} \tag{1}$$

where $\mathbf{R_{ab}}, \boldsymbol{t}_{ab}$ are rotation and translation of cameras C^b wrt. C^a, and $\boldsymbol{n}$ and d are orientation and distance of the underlying plane wrt. C_a which maps points according to $\bar{x_b} = \mathbf{H_{ab}}\bar{x_a}$. Thus, if $\mathbf{H}$ can be estimated precisely, the external camera parameters are known (up to scale which is encoded in d). Furthermore, any plane visible in both cameras can be used to estimate $\mathbf{R_{ab}}, \boldsymbol{t}_{ab}$, just $\boldsymbol{n}$ and d are plane-dependent.

For our scenario, an intuitive idea would be to select the foot points as $\bar{x}$ from four pedestrians in two views and compute $\mathbf{H}$ (see Fig 2c). However, the planarity might not be valid and, more important, all foot point correspondences have to be true which is unlikely for large baselines with low precision. Formally, this plane configuration would be four inter-camera correspondences, one time instance (intra-camera correspondence), and two cameras, or $\mathcal{C}_{41} = (4, 1, 2)$ as done in [21]. Since $\mathbf{H_{ab}}$ has eight degrees of freedom, the product of the configuration elements has to be 8. Since the number of cameras is 2, the configuration space is quite limited, so only $\mathcal{C}_{22} = (2, 2, 2)$ and $\mathcal{C}_{14} = (1, 4, 2)$ would be alternative minimal sampling sets.

Out of these, $\mathcal{C}_{14}$ (Fig 2a) also intuitively aims at determining the ground plane, but this time from one point visible over time. Compared to $\mathcal{C}_{41}$, it has the advantage that only one true inter-camera correspondence is needed. However, the configuration is degenerate if the four intra-camera correspondences are on on a line, which often occurs during human walking.

We focus on the $\mathcal{C}_{22}$ configuration, which means two points in two frames for two cameras (Fig. 1 and Fig 2b). Compared to C_{14}, it has the disadvantage that two inter-camera correspondences are needed. However, by establishing correspondences between pedestrians instead of points, with stably-localizable head and foot points, we only need *one* inter-camera pedestrian correspondence and *two* time instances where it is seen.

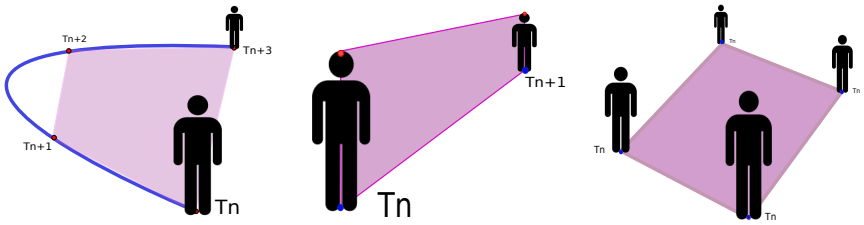

Fig. 3: Different configurations for $\mathbf{H}$ estimation (magenta plane). Fig 2a shows $\mathcal{C}_{14} = (1, 4, 2)$ an example of a pedestrian walking (blue track) from which we select four points. Fig 2b $\mathcal{C}_{22} = (2, 2, 2)$ is the situation used in this paper, with two points extracted at two timesteps and Fig 2c presents $\mathcal{C}_{41} = (4, 1, 2)$ an estimation of a ground plane using four foot points at the same timestep.

4 Markov chain Monte Carlo Cascade for likelihood maximization

The presented algorithm takes as inputs m synchronized sequences produced by cameras C^i forming a network N. We assume the intrinsic calibration K^i known. For automatic temporal synchronization, audio signals could be cross-correlated, and for intrinsic parameters, automatic techniques, as presented in section 2, could be used.

These sequences will be in first place pre-processed to extract pedestrian tracklets. These tracklets will be used to produce the so-called foot head points as explained in section 4.1, similarly to [14, 20, 12].

As a second step, tracklets in different views will be associated in pedestrian pair hypotheses. These hypotheses will be the basic elements of the pairwise pose estimation process (section 4.2) which aims at estimating the relative pose density for all camera pairs. This will be conducted as a Markov chain Monte Carlo procedure which follows the idea of [14]. Once this density is estimated for every camera pair, we will fuse the pairwise pose estimates into a consistent network by a triangular structural term (section 4.3).

Our algorithm is a 3-layer cascade of Markov chain Monte Carlo sampling. Each layer approximates different posterior densities and uses their estimations to feed on the next layer. The first layer is at the pedestrian observation level, which will suggest relative poses for a second layer at a camera pair level. We formulate this global network optimization as a maximum likelihood problem which will be solved by a last layer of sampling.

4.1 Pre-Processing

The first step is to detect and track pedestrians to produce pedestrian tracklets. For detection, we use a standard deformable part model as presented in [10] and used the already trained algorithm provided online and trained on the Pascal VOC dataset [22]. Having the detection bounding boxes, we create the tracklets using the flow algorithm

proposed in [11] and kindly distributed online by the authors. Please note that the tracklets might be incomplete or falsely developed. Our approach is designed to simply not consider these for the calibration in a later steps. The j^{th} tracklet in camera C^i is noted t^i_j. For computational purpose we filter out all short tracklets with less than 2 seconds overlap.

Based on these tracklets, we will extract the commonly called foot head points for every pedestrian at every frame they are seen. To extract such points in a frame f, we use a naive background subtraction method in the form of a grayscale per-pixel median filter. Albeit being naive and extremely simple, this background filtering method has been sufficient. We then compute the principal components of the foreground enclosed inside the detection bounding box. As humans are mainly distributed along the vertical axis, itself colinear to the gravity vector, we take the first principal axis $\boldsymbol{v}_{pa}$ and the center of mass of the foreground $\boldsymbol{p}_{cm}$ to extract two points using it. The head point is simply computed as $\boldsymbol{p}_{cm} + \boldsymbol{v}_{pa}$ and the corresponding foot point $\boldsymbol{p}_{cm} - \boldsymbol{v}_{pa}$. This naive point extraction method is motivated by the will of having a robust algorithm as accurate points or tracklets may not be available in dynamic environment where lots of occlusions can occur.

4.2 Pairwise Pose Density Estimation

We will first derive the estimation of the probability density of a relative pose $(\mathbf{R}, \boldsymbol{t})$ between a camera pair (C^a, C^b), given all correspondences in-between:

$$p_{\text{cc}}(\mathbf{R}, \boldsymbol{t} \mid C^a, C^b) \tag{2}$$

In this section, we assume that there is no side information from the network. p_{cc} is modeled as Parzen density over previous estimates of $\mathbf{R}, \boldsymbol{t}$. In order to iteratively refine p_{cc}, we sample a correspondence from all hypotheses which, in turn, will be used to compute another $\mathbf{R}', \boldsymbol{t}'$ sample, as explained later in section 4.2. To guide multinomial correspondences sampling, we use p_{cp} as described in the following. We apply Bayes' theorem to transform (2) into:

$$p_{\text{cc}}(\mathbf{R}, \boldsymbol{t} \mid C^a, C^b) = \frac{p_{\text{cp}}(C^a, C^b \mid \mathbf{R}, \boldsymbol{t}) p(\mathbf{R}, \boldsymbol{t})}{p(C^a, C^b)} \tag{3}$$

The denominator $p(C^a, C^b)$ is usually considered constant and we do the same for $p(\mathbf{R}, \boldsymbol{t})$ which would be a prior on the relative pose. We devise p_{cp} as:

$$p_{\text{cp}}(C^a, C^b \mid \mathbf{R}, \boldsymbol{t}) = \prod_{(t^a_i, t^b_j) \in (C^a, C^b)} p_{\text{pp}}(t^a_i, t^b_j \mid \mathbf{R}, \boldsymbol{t}) \,, \tag{4}$$

Following is the definition of the pedestrian pair probability with the reprojection error modeled by the Blake-Zisserman distribution:

$$p_{\text{pp}}(t^a_i, t^b_j \mid \mathbf{R}, \boldsymbol{t}) = \sqrt[(\delta d)^\phi]{\prod_{f=1}^{\delta d} e^{-\frac{e(t^a_i, t^b_j, f \mid \mathbf{R}, \boldsymbol{t})^2}{2\sigma^2}}} + \epsilon, \tag{5}$$

where $e(t_i^a, t_j^b, f \mid \mathbf{R}, \boldsymbol{t})$ is the reprojection error between t_i^a and t_j^b, evaluated at the foot head points of frame f. δd denotes the tracklet length. $\phi \in [0, 1]$ is an independence parameter. As we expect our pedestrian pairs to be correlated, we used $\phi = 0.5$ in our experiments. $\epsilon = 0.01$ represents the uniform noise in the data. The reprojection standard deviation σ has been set to a standard value of 5 pixels.

The reprojection error e at frame f is defined as average error of the respective head and foot points in both cameras C^a and C^b. To evaluate e for a head or foot point correspondence $(\boldsymbol{x}^a, \boldsymbol{x}^b)$, we use $\mathbf{R}$ and $\boldsymbol{t}$ to triangulate $(\boldsymbol{x}^a, \boldsymbol{x}^b)$ to the 3D point $\boldsymbol{X}$. After re-projecting $\boldsymbol{X}$ into both camera views, the Euclidean distance to $\boldsymbol{x}^a$ and $\boldsymbol{x}^b$ is used for e.

Relative Pose Guided Sampling Having sampled a particular pair of tracklets (t_i^a, t_j^b) in the previous section, it will be used to generate the next relative pose $\mathbf{R}', \boldsymbol{t}'$ by sampling with respect to:

$$\begin{aligned} p_{\text{rt}}(\mathbf{R}, \boldsymbol{t} | t_i^a, t_j^b) &= \frac{p_{\text{pp}}(t_i^a, t_j^b \mid \mathbf{R}, \boldsymbol{t}) p(\mathbf{R}, \boldsymbol{t})}{p(t_i^a, t_j^b)} \\ &\propto p_{\text{pp}}(t_i^a, t_j^b \mid \mathbf{R}, \boldsymbol{t}) p_{\text{cc}}(\mathbf{R}, \boldsymbol{t} \mid C^a, C^b) \\ &\propto p_{\text{pp}}(t_i^a, t_j^b \mid \mathbf{R}, \boldsymbol{t}) p_{\text{cp}}(C^a, C^b \mid \mathbf{R}, \boldsymbol{t}) \end{aligned} \tag{6}$$

The theorem of Bayes is used here again, we assume the denominator constant again. Note that you could also use it as a prior on the pedestrian pair correspondence from other information (e.g. an appearance prior). We incorporated the prior $p(\mathbf{R}, \boldsymbol{t}) = p_{\text{cc}}(\mathbf{R}, \boldsymbol{t} \mid C^a, C^b) \propto p_{\text{cp}}(C^a, C^b \mid \mathbf{R}, \boldsymbol{t})$ (thanks to (3)) as an indication of the overall likelihood of a relative pose with respect to the camera pair.

Density p_{rt} will also be modeled as a Parzen density of the previously visited $\mathbf{R}, \boldsymbol{t}$. Every time correspondences are selected to suggest a new relative pose, they firstly produce a completely new $\mathbf{R}, \boldsymbol{t}$ as explained in section 3 as an exploration step and incorporate it into the Parzen density estimate. Then $\mathbf{R}', \boldsymbol{t}'$ is sampled from p_{rt}. This procedure allows us to balance exploration and exploitation with the goal to find likely solution for (3).

4.3 Network Configuration Optimization

We now have a description of (3) for every pair of cameras (C^a, C^b). We want to combine them to recover the structure of the network, in other words we want to find the most likely set of relative poses for each edge that produces a consistent network. By consistent we mean relative poses that produces triangle stable network. We define triangle stability by:

$$e_\Delta = \| I_d - P^{ab} \circ P^{bc} \circ P^{ca} \|_f \tag{7}$$

Where I_d is the identity matrix, $\| \cdot \|_f$ the Frobenius norm [23] and P^{ij} are 4x4 pose matrices from C^i to C^j. The smaller e_Δ is, the more consistent the triangle is. Due to the unknown scale that exists between the relative poses, we cannot directly concatenate them. To overcome this problem we locally solve for the scale, each of the three relative

poses brings an unknown scale, we fix one to 1 and use a least square solution to obtain the two left. After rescaling the translation component of the relative poses, they can be used for comparison.

One could just select the most likely relative pose for each camera pair but it would probably violate the triangle constraints ($e_\Delta \gg 0$). We formalize this problem as a maximization for the relative pose set $\mathcal{P} = \{P^{ab} \forall (C^a, C^b)\}$ which fulfills camera pairs likelihood p_{cp} and network constraints p_Δ:

$$\begin{aligned} arg\,max_{\mathcal{P}} p(N \mid \mathcal{P}) = &\prod p_{\text{cp}}(C^a, C^b \mid P^{ab}) \\ &\cdot \prod p_\Delta(P^{ab}, P^{bc}, P^{ca}) \end{aligned} \tag{8}$$

The first product of p_{cp} is our data term coming directly from the previous step (equation 4) and reflecting the pedestrian observations. The second product is a structural term based on e_Δ which is modeled as a Gaussian:

$$p_\Delta(P^{ab}, P^{bc}, P^{ca}) = e^{-\frac{\|I_d - P^{ab} \circ P^{bc} \circ P^{ca}\|_f^2}{2\beta^2}} \tag{9}$$

We empirically found $\beta = 0.25$ for good results. Setting it too low blocks the exploration into local minima and too high does not guide the sampling anymore.

4.4 Gibbs Metropolis Hastings Sampling

We now show how we can maximize (8) using a random walk algorithm. Due to nonlinearity and high dimensionality, it is extremely hard to solve this maximization problem by a direct approach or exhaustive testing. We propose to use the Gibbs sampling approach tinted with Metropolis Hastings acceptance ratio to walk through the state space. Our Gibbs sampling approach creates a network state vector S^N of $n = {}^1\!/_2 \cdot m(m-1)$ (m = number of cameras in N) random variables P^{ab}, each corresponding to the relative pose of one camera pair. At every iteration, every random variable P^{ab} of the network state vector S^N is updated by sampling it from the distribution

$$p(P^{ab} \mid S^N \setminus \{P^{ab}\}) \tag{10}$$

In our work, we composed this distribution using the camera pair data term (3) and a product of the triangle stability term (9) which leads the sampling around locations that produce consistent network configurations:

$$\begin{aligned} p(P^{ab} \mid S^N \setminus \{P^{ab}\}) = &p_{\text{cc}}(P^{ab} \mid C^a, C^b) \\ &\cdot \prod_{i \notin \{a,b\}} p_\Delta(P^{ab}, P^{bi}, P^{ia}) \end{aligned} \tag{11}$$

Sampling from this distribution gives us a new state $P^{ab\prime}$. In order to guide this sampling more strongly towards the optimal solution, we spice the standard Gibbs sampling by computing an acceptance ratio α as follows:

$$\alpha = \frac{p(N \mid S^{N'})}{p(N \mid S^N)} \tag{12}$$

Where $S^{N\prime}$ is the network state vector where P^{ab} has been replaced by $P^{ab\prime}$. The actual state vector S^N is updated based on the value of α. If α is bigger than one, which would mean accepting this new state increases the probability of having a correct network, we accept the change. Otherwise it means the change will decrease the quality of the current estimate. In this case, we accept the change proportionally to α (small decrease in quality have more chances to be accepted than big ones). This process is summarized in Algorithm 1

Data: Discrete estimation of densities $p_{\text{cc}}(P^{ab} \mid C^a, C^b)$
Result: Best network configuration S^B
$\forall\,(C^a, C^b) : P^{ab} \leftarrow arg\,max p_{cp}(P^{ab} \mid C^a, C^b)$;
$S^N \leftarrow \{P^{ab}\}$;
$S^B \leftarrow S^N$;
for *n_iterations* **do**
  for $\forall\, P^{ab} \in S^N$ **do**
    $P^{ab\prime} \leftarrow$ sample from $p(P^{ab} \mid S^N \setminus \{P^{ab}\})$;
    $\alpha \leftarrow \frac{p(N|S^{N\prime})}{p(N|S^N)}$;
    if *rand()* $< \alpha$ **then**
      $S^N \leftarrow S^N \setminus \{P^{ab}\} \cup P^{ab\prime}$;
    end
    if $p(N \mid S^N) > p(N \mid S^B)$ **then**
      $S^B \leftarrow S^N$;
    end
  end
end

Algorithm 1: Outline of the global network optimization sampling algorithm.

By the random walk, we explore the density in equation (8). Likely solutions are sampled preferably, but to overcome local minima, unlikely solutions are also explored. As final network configuration result, we use the most-probable explored state according to (8).

4.5 Smallest Stable Triangular Spanning Tree

As not all cameras are connected with each other (no common pedestrians observations) and some estimates are very hard (unstable camera pair configuration), we add a final selection step that will select only the best relative poses. Indeed, we computed the relative pose for every camera pair, yet we only need a subset of it in order to be able to calibrate it up to one unknown scale. We used an augmented version of the shortest triangle paths algorithm presented by [24, 25]. This algorithm produces triangle paths which are triangular connected, meaning it is enough for estimating all the unknown scales down to one global scale.

For the sake of conciseness, we refer to [25], and briefly explain our extensions: we incorporate our triangle stability probabilities combined with our pairwise likelihoods

as edge weights on top of their graphical model. This allows us to find paths that correctly explain pedestrian observations as well as having a structurally stable network. We finally differ from their approach by initializing the algorithm from multiple different entry nodes instead of the one with highest probability, and by selecting the most probable paths set as solution.

5 Experimental Results

Fig. 4: The first images of all 7 cameras of the PETS 2009 sequence.

As input data, we take the sequences PETS 2006 S1-T1-C and 2009 S2.L1 (Fig. 4) consisting of 3'021 frames per 4 cameras, and 795 frames for each 7 cameras respectively. The cameras record a central scene from 360 degrees. As evaluation metric, we use relative camera pose differences in percentage to the groundtruth. For rotation comparison, between estimate $\mathbf{R}$ and ground truth $\mathbf{R}'$, we based our distance measure on [26] and used the $\boldsymbol{\Phi}_5$ distance function $|\mathbf{I_d} - \mathbf{R}'\mathbf{R}^\intercal|_\mathrm{f}$. As scale invariant translation measure, we compute the angular difference between the translation vectors.

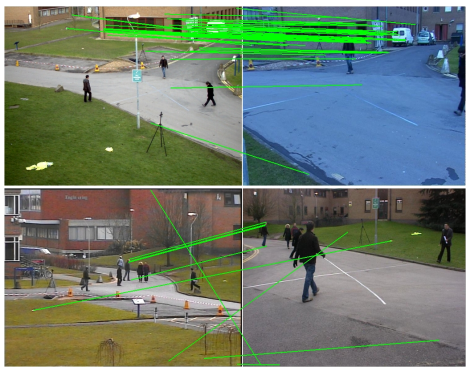

Fig. 5: Top: Only frame pair of PETS2009 with an inlier-only correspondence set. Bottom: Typical failure containing overfitted outliers.

5.1 Structure-for-Motion as Baseline Comparison

To demonstrate the difficulty of calibrating these sequences, we test the state-of-the-art structure-from-motion pipeline from VisualSfM [4, 5] to compensate for it not be-

ing able to take as input grountruth intrinsic calibration, we guided its estimation by providing the ground truth in the EXIF information of the image. Manual verification of estimated intrinsic proved it to be correctly estimated. Firstly, a standard Structure-from-Motion is conducted. The first images of all cameras are matched and their relative pose is verified with an epipolar model. Although we tried several ways of optimizing the results, out of the 21 relative inter-camera poses, only one was estimated from inlier correspondences (Fig. 5) leading to early breakdown of the algorithm.

We then proceeded to use the same input for VisualSfM as our algorithm uses. The complete list of corresponding foot head points for every hypothesis is fed to VisualSfM. Unfortunately, due to the number of outliers dominating the inliers, only a small subset of the cameras are estimated. Multiple repetitions have been conducted for the calibration and the best, selected firstly on the number of cameras then on the actual error measurements, is shown in Table 1. In order to provide a comparison baseline,

Table 1: Baseline results produced using VisualSfM with the same input as our algorithm.

Dataset	Pets2009 S2.L1		Pets2006 S1-T1-C	
Remarks	Only **4** cameras		Only **2** cameras	
	$\mathbf{R}$	$\boldsymbol{t}$	$\mathbf{R}$	$\boldsymbol{t}$
Mean	15.0	35.8	7.1	4.8
Std	9.5	28.3	0	0
Min	3.4	7.4	7.1	4.8
Max	21.2	64.2	7.1	4.8

we added the pedestrian correspondence knowledge for VisualSfM and used only foot head points of true manually annotated pedestrian correspondences as input. Again only partial networks are estimated. The results are summarized in Table 2

Table 2: Baseline results produced using VisualSfM and pedestrian correspondence knowledge.

Dataset	Pets2009 S2.L1		Pets2006 S1-T1-C	
Remarks	Only **5** cameras		Only **3** cameras	
	$\mathbf{R}$	$\boldsymbol{t}$	$\mathbf{R}$	$\boldsymbol{t}$
Mean	8.9	5.3	5.9	2.7
Std	4.4	1.1	2.2	2.0
Min	2.6	3.2	4.6	0.6
Max	18.0	6.8	8.5	4.6

5.2 Camera Calibration using our Cascade of Markov-Chain Monte-Carlo

To evaluate our algorithm, we compare our estimation to the dataset groundtruths. Two different scenarios are presented, the first one named *whole* network will compare every estimated relative poses whereas *minimal* network will evaluate only the minimal

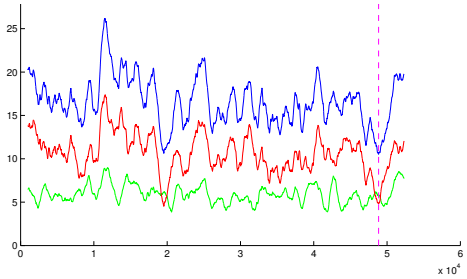

Fig. 6: Plot of energy during a global optimization. Red shows the pairwise energy, green the network energy and blue the total energy. The minimal energy is denoted by the magenta line. (Energies have been smoothed for readability purposes).

relative pose set computed using the triangle path algorithm from section 4.5. The pairwise pose estimation process is iterated for 5000 iterations and the global network optimization is conducted for 2500 iterations. On non-optimized Matlab code, the pairwise density estimation took in average 1 hour per camera pair and the global network optimization a few hours. We believe it to be optimizable and parallelizable.

Table 3 presents the error measures for Pets2009 and Pets2006 datasets.

We can see that the triangle path extraction allows us to increase the quality of the network on every aspect. Note that Pets2006 dataset has only four cameras resulting in six different camera pairs, when the *minimal* network solution needs five camera pairs to cover the whole network, limiting the selection of better relative poses and thus the amelioration in results. By comparing the *minimal* columns of Table 3 and Table 3 to Table 2, we can see that our algorithm is able to correctly estimate relative poses with an error similar to what our baseline with groundtruth correspondences knowledge is able to achieve, however it is to be noted that our approach produces estimates for every camera. Also note that our algorithm estimates all cameras when the baseline for Pets2009 only managed to produce estimates for 5 cameras.

As the presented algorithm belongs to the random walk algorithm family, it is interesting to look at the energy variation as shown in Fig 6. It can be seen that the global minimum may not be at the minimum of either the pairwise or network energies and sometimes accepting worse relative poses can lead to improving the overall solution by avoiding local minima.

Table 3: Rotational error relative to groundtruth as percentage to the grountruth.

Dataset	Pets2009 S2.L1				Pets2006 S1-T1-C			
Network	*whole*		*minimal*		*whole*		*minimal*	
	$\mathbf{R}$	$\boldsymbol{t}$	$\mathbf{R}$	$\boldsymbol{t}$	$\mathbf{R}$	$\boldsymbol{t}$	$\mathbf{R}$	$\boldsymbol{t}$
Mean	15.7	8.1	10.5	6.1	13.2	6.6	10.3	5.7
Std	10.7	6.5	5.5	4.0	6.9	5.0	7.3	5.2
Min	2.4	0.6	2.4	0.6	3.4	0.8	3.4	0.8
Max	37.5	23.9	20.3	14.4	21.6	13.6	19.0	10.7

A collateral result from network external calibration is the possibility to match pedestrians in different views. We computed the Jaccard index between inlier pedestrian set produced by the groundtruth pose and our estimated pose. We achieve in average 31.3% on the Pets2009 sequence and 25.6% on the Pets2006. Unfortunately, pedestrian correspondence are not found reliably when the relative pose estimation error is too big (when $< 5\%$, the average Jaccard index is 64.6% but when $> 15\%$ it becomes 0%) but remarkably, the triangle stability term allows the use of false correspondences for calibration. Figure 8 shows some typical falsely found correspondences: reasonable results but false due to occlusion, and overfitting during the triangular reprojection.

In a last experiment, we took pedestrian correspondence estimates (pedestrian estimated with likelihood above 1% as shown in Figure 7) and used their foot head points as input in VisualSfM to obtain a robust relative pose estimation. This is the same process as for our baseline, except we do not take the inlier pedestrians from the groundtruth but from our estimation, to see if our algorithm is able to produce such knowledge accurately. The results are shown in Table 4. In Pets2009 sequence, only a subset of cameras managed to be correctly estimated with errors in the same range as our estimates and the baseline. Pets2006 produced no correct results for our estimates due to too many wrong estimated correspondences. It can be seen that non-linear optimizations do not improve our estimates significantly.

Table 4: VisualSfM results as a post processing over our pedestrian inlier estimation for Pets2009.

Dataset	Pets2009 S2.L1	
Remarks	Only **4** cameras	
	$\mathbf{R}$	$\boldsymbol{t}$
Mean	9.6	5.8
Std	5.5	2.3
Min	3.5	0.8
Max	18.0	6.5

6 Conclusion & Future Work

We have presented a probabilistically justified algorithm in the form of a cascade of Markov chain Monte Carlo algorithm and applied it to the task of estimating the external calibration of a camera network by observing unconstrained pedestrians. This algorithm has many advantages, its probabilistic formulation providing it an adaptability property for whoever would like to incorporate more priors (e.g. appearance prior). A collateral result of a good estimation is the pedestrian correspondences deduction which can be useful in many surveillance situations and used as prior information for other processes. Its main quality is its robustness against noisy data due to inexact point extraction or drifting tracklets and against outlier dominated hypotheses. Lastly, we think the simplicity of this algorithm allows it to be improved on different aspects such as the relative pose estimation in itself but also by addressing wider problems such as camera synchronization or intrinsic calibration.

Fig. 7: Inlier-correspondence examples after our optimization. Magenta and yellow lines show respectively corresponding head and foot points. First row shows Pets2009 sequence and the bottom one: Pets2006.

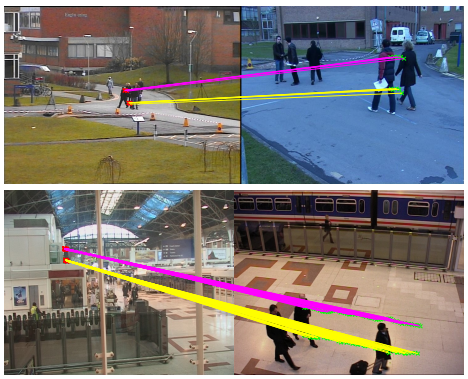

Fig. 8: Typical false correspondence examples after our optimization, labeled as in Figure 7.

Acknowledgement. This research was supported by the SNF project "Tracking in the Wild" CRSII2_147693/1.

References

1. Svoboda, T., Martinec, D., Pajdla, T.: A convenient multi-camera self-calibration for virtual environments. PRESENCE: Teleoperators and Virtual Environments **14** (2005) 407–422
2. Zhang, Z.: A flexible new technique for camera calibration. IEEE Trans. Pattern Anal. Mach. Intell. **22** (2000) 1330–1334
3. Baker, P., Aloimonos, Y.: Complete calibration of a multi-camera network. In: Omnidirectional Vision, 2000. Proceedings. IEEE Workshop on. (2000) 134–141
4. Wu, C.: Towards linear-time incremental structure from motion. In: Proceedings of the 2013 International Conference on 3D Vision. 3DV '13, Washington, DC, USA, IEEE Computer Society (2013) 127–134

5. Wu, C., Agarwal, S., Curless, B., Seitz, S.M.: Multicore bundle adjustment. In: In IEEE Conference on Computer Vision and Pattern Recognition (CVPR, IEEE (2011) 3057–3064
6. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence **27** (2005) 1615–1630
7. Junejo, I.N.: Using pedestrians walking on uneven terrains for camera calibration. Mach. Vis. Appl. **22** (2011) 137–144
8. Chen, T., Bimbo, A.D., Pernici, F., Serra, G.: Accurate self-calibration of two cameras by observations of a moving person on a ground plane. In: AVSS, IEEE Computer Society (2007) 129–134
9. Liu, J., Collins, R.T., Liu, Y.: Robust autocalibration for a surveillance camera network. IEEE Winter Conference on Applications of Computer Vision **0** (2013) 433–440
10. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Transactions on Pattern Analysis and Machine Intelligence **32** (2010) 1627–1645
11. Hamed Pirsiavash, Deva Ramanan, C.F.: Globally-optimal greedy algorithms for tracking a variable number of objects. Computer vision and Pattern Recognition CVPR (2011)
12. Rother, D., Patwardhan, K.A., Sapiro, G.: What can casual walkers tell us about a 3d scene? In: ICCV, IEEE (2007) 1–8
13. Krahnstoever, N., Mendona, P.R.S.: Autocalibration from tracks of walking people. In: in Proc. British Machine Vision Conference (BMVC. (2006) 4–7
14. Krahnstoever, N., Mendona, P.R.S.: Bayesian autocalibration for surveillance. In: ICCV, IEEE Computer Society (2005) 1858–1865
15. Micusik, B., Pajdla, T.: Simultaneous surveillance camera calibration and foot-head homology estimation from human detections. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. (2010) 1562–1569
16. Kusakunniran, W., Li, H., 0002, J.Z.: A direct method to self-calibrate a surveillance camera by observing a walking pedestrian. In: DICTA, IEEE Computer Society (2009) 250–255
17. Lv, F., Zhao, T., Nevatia, R.: Self-calibration of a camera from video of a walking human. In: ICPR (1). (2002) 562–
18. Lv, F., Zhao, T., Nevatia, R.: Camera calibration from video of a walking human. Pattern Analysis and Machine Intelligence, IEEE Transactions on **28** (2006) 1513–1518
19. Liu, J., Collins, R.T., Liu, Y.: Surveillance camera autocalibration based on pedestrian height distributions. In: British Machine Vision Conference (BMVC). (2011)
20. Possegger, H., Rther, M., Sternig, S., Mauthner, T., Klopschitz, M., Roth, P.M., Bischof, H.: Unsupervised calibration of camera networks and virtual ptz cameras. In: Proc. Computer Vision Winter Workshop (CVWW). (2012)
21. Puwein, J., Ziegler, R., Ballan, L., Pollefeys, M.: Ptz camera network calibration from moving people in sports broadcasts. In: WACV, Breckenridge, Colorado (2012)
22. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International Journal of Computer Vision **88** (2010) 303–338
23. Golub, G.H., Van Loan, C.F.: Matrix Computations (3rd Ed.). Johns Hopkins University Press, Baltimore, MD, USA (1996)
24. Bajramovic, F., Denzler, J.: Global uncertainty-based selection of relative poses for multi camera calibration. In: Proceedings of the British Machine Vision Conference, BMVA Press (2008) 74.1–74.10 doi:10.5244/C.22.74.
25. Bajramovic, F., Brückner, M., Denzler, J.: An efficient shortest triangle paths algorithm applied to multi-camera self-calibration. Journal of Mathematical Imaging and Vision (JMIV) (2011) 1–14
26. Huynh, D.Q.: Metrics for 3d rotations: Comparison and analysis. J. Math. Imaging Vis. **35** (2009) 155–164