Camera Calibration by Global Constraints on the Motion of Silhouettes

Gil Ben-Artzi

Weizmann Institute of Science
Rehovot, Israel

Intel Labs
Santa Clara, CA, USA

Abstract

We address the problem of epipolar geometry using the motion of silhouettes. Such methods match epipolar lines or frontier points across views, which are then used as the set of putative correspondences. We introduce an approach that improves by two orders of magnitude the performance over state-of-the-art methods, by significantly reducing the number of outliers in the putative matching. We model the frontier points' correspondence problem as constrained flow optimization, requiring small differences between their coordinates over consecutive frames. Our approach is formulated as a Linear Integer Program and we show that due to the nature of our problem, it can be solved efficiently in an iterative manner. Our method was validated on four standard datasets providing accurate calibrations across very different viewpoints.

1. Introduction

Multi-camera systems are becoming increasingly more popular for $3D$ reconstruction, marker-less motion capture, surveillance, and even for ”smart homes”. Traditionally, epipolar geometry is computed by finding the corresponding points between cameras. However, in such a setting many camera pairs are from very different viewpoints and consequently, not enough reliable feature points can be matched automatically. In cases where the silhouettes of the objects in the scene are available or are easily extracted, it has been proposed to use their motion for calibration [34]. These methods match tangent epipolar lines on the convex hull of the silhouettes across views [33]. For each frame, a similarity function for every possible matching of the corresponding tangent epipolar lines is evaluated and the pair with the highest similarity is selected as a putative correspondence [7]. Recovering the corresponding tangent epipolar lines induces matching between the projections of special points, denoted as *extremal frontier points*, across views where occlusions do not occur (see [19, 33, 12, 28] for a detailed description). Hereafter, when referring to frontier points, we will refer to the extremal $3D$ frontier points or to the image points that represent their projections. The epipolar geometry is evaluated by using RANSAC [16] with the putative list of matching epipolar lines or frontier points.

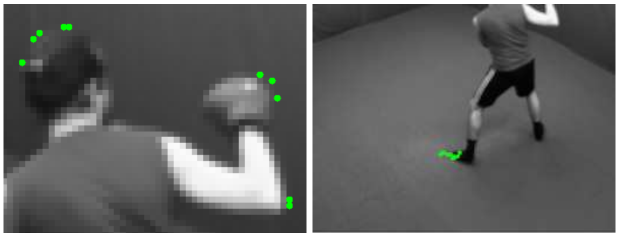
(a) (b)

Figure 1. (a) The set of possible frontier points are denoted by green circles. It is difficult to distinguish between true and false frontier points because often the candidates are at nearby positions and share the same tangents to the convex hull. (b) The trajectory of the true frontier points over 15 consecutive frames. We look for frontier points whose positions vary slowly over time.

A key limitation of previous approaches is the high number of outliers in the recovered list of corresponding epipolar lines or frontier points. This is mainly because two nearby points on the convex hull might make only a minor difference in their associated tangents. Here, we present an approach that markedly reduces the number of outliers. This leads to an improvement of two orders of magnitude in the performance. Unlike previous approaches where the putative matching is carried out for each frame independently, we require that the positions of frontier points in consecutive frames will vary slowly. Fig. 1.a shows a set of possible frontier points in the image denoted by full green circles. Since nearby points share similar tangents, it is hard to recover the true frontier points. Fig. 1.b shows the trajectories of the true frontier points over 15 frames. By looking for frontier points that vary slowly over time, we can accurately recover their positions, up to sub-pixel accuracy.

We model the problem as a constrained flow optimization problem. At each time instant we look for two corresponding pairs of frontier points. This translates into finding two paths in the graph under a set of constraints. The graph is formulated as an integer programming (IP) problem whose global maximum can be recovered with reasonable complexity. However, since there is a large number of variables and constraints, the solution to this problem might be slow and not scalable for long sequences. We therefore provide an efficient algorithm for obtaining an approximate solution by iteratively applying the shortest path algorithm. We show that for any practical camera setting, the global optimum is recovered.

This paper therefore contributes by presenting: (a) a reformulation of the problem of finding frontier points across views as a constrained graph optimization problem, (b) a practical algorithm to recover its global maximum, and (c) an accurate calibration method with superior performance over state-of-the-art methods. We validated our approach on several standard datasets and show that it is highly effective.

2. Related Work

The most common uses of silhouettes in multi-camera systems are for shape-from-silhouettes [11, 17, 3] and camera calibration [23, 33, 10, 32, 39]. In shape-from-silhouettes, the goal is to recover the visual hull [27, 30] of the object. In calibration, it is assumed that the motion of the silhouette is fully observed across different views. Correspondences are established between frontier points across different views based on the epipolar tangency constraints [12] and are then used to compute the epipolar geometry. Most methods require a specific configuration that cannot be applied in a general setting, which is considered here. These include calibrated cameras, static objects, orthographic projection models, or known (turntable) motion [18, 20, 37, 29, 23]. Calibration can be carried out without explicitly finding the corresponding tangent epipolar lines, such as in [10, 38]. These methods require a good initial guess to converge.

In this paper we consider calibration in the most general setting where only the motion of silhouettes is available without further assumptions. Sinha and Pollefeys [33] considered such a setting for calibration. They searched for the epipoles by randomly sampling lines from the tangent envelope of the silhouette. They used RANSAC for extracting the most plausible solution. Ben-Artzi et al.[7] proposed a temporal binary descriptor, denoted as a motion-barcode, for suggesting the corresponding epipolar lines across views. They sampled the lines from the tangent envelope according to the similarity induced by their descriptor. Importantly, they showed that accuracy and runtime are markedly improved. In both methods there is a significant number of outliers and the frontier points in the current frame are matched without taking into account the previous or the next corresponding points. Kasten et al.[26] used the same descriptor for calibrating crowded scenes. However, they considered noisy, low-resolution images and provided limited accuracy.

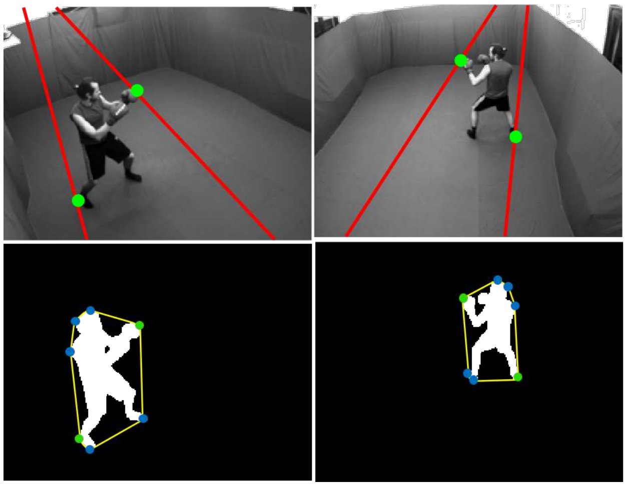

Figure 2. (a) Top row. Two corresponding images captured from different views. The red lines are the epipolar lines. The green circles are the ground truth frontier points. (b) Bottom row. The frontier points can only be derived from a small set of points. These are points on the convex hull intersecting with the silhouette; they are denoted by a circle and are termed *critical points*. The green critical points are also the ground truth frontier points.

We used the similarity proposed by the binary motion-barcode descriptor [7] as one of the cues for our algorithm. Similar motion-based binary descriptors have been proposed and used by [15],[14]. Both methods assume a planar structure of the scene. Ben-Artzi et al. [8] used a similar descriptor for matching events across different views, but their method does not provide accurate localization. Pundik and Moses [31] introduced a similar motion-based descriptor, line signal, and used it for video synchronization. It depends on the color and was used under the assumption of known calibration.

3. Our Model

We assume that we have two sequences of binary silhouette images, each captured from a different view. Consider the image captured from the first view as the left image and the image captured from the other view as the right image. In each of the images, the convex hull of the silhouette is extracted and the subset of points on the intersections of the convex hull and the silhouettes are identified. These points are termed *critical points*. The true correspondence of the extremal frontier points across views can only be found between critical points [13, 33], such as is illustrated in Fig. 2. In addition, we assume that we are given a similarity measure for the correspondence of critical points across views.

We model the matching problem as a constrained flow in a graph. Our goal is to recover the two paths that maximize

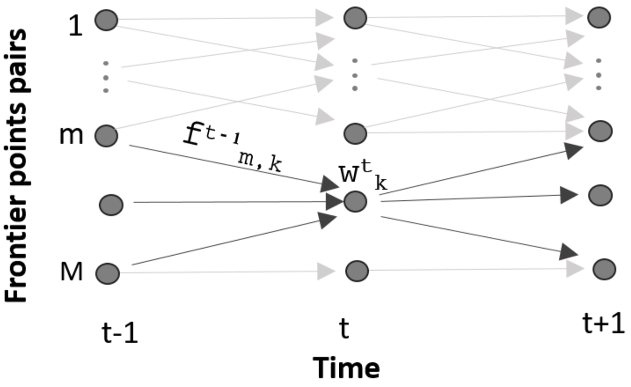

Figure 3. Our problem is represented as a directed acyclic graph (DAG). Every vertex in graph v_t^k is associated with a binary variable w_k^t and every edge in graph $e^{t-1}_{m,k}$ is associated with a binary variable $f^{t-1}_{m,k}$. The vertices represent all possible pairs of matched frontier points across views.

the flow. Each path is a sequence of corresponding pairs of frontier points across views, at each time instant.

3.1. Formulation

We introduce a directed acyclic graph (DAG) $G = (V, E)$. Let x_i^t denote a critical point in the left image and $x_j'^t$ denote a critical point in the right image, both at time instant t. Each vertex v_k^t in the graph represents a pair of two critical points $(x_i^t, x_j'^t)$. Each edge between vertices represents an admissible transition between a true match at time t and a true match at time $t+1$. In our case we consider all possible transitions between the time instants as admissible transitions. The number of vertices at each time instant is $M = K_1 \times K_2$ vertices, where K_1 and K_2 are the total number of critical points in the left and right images, respectively. Let $N(k) \subset \{1...K\}$ be the neighborhood of vertex k. There is an edge $e_{i,j}^t$ from v_i^t to v_j^{t+1} if and only if $j \in N(i)$. Every vertex is associated with a binary variable $w_i^t \in \{0, 1\}$, indicating whether the current pair of critical points are a true match of frontier points. Each edge $e_{i,j}^t$ is associated with a binary variable $f_{i,j}^t$, indicating whether both v_i^t and v_j^{t+1} are a true match. Such a graph is illustrated in Fig .3.

We now define the set of constraints that will ensure that the set of flows accurately represent our matching problem. First, for each vertex in the graph, the flow conservation implies that the sum of flows coming into a vertex at time $t-1$ equals the sum of flows outgoing from the vertex at time t:

$$\forall t, j \sum_{i \in N(j)} f_{i,j}^{t-1} = \sum_{k \in N(j)} f_{j,k}^t = w_j^t. \tag{1}$$

We refer to $N(x)$ as either edges outgoing from vertex x or incoming into vertex x, depending on the meaning of the index x.

Second, in our formulation, each vertex is either a true or false match between frontier points. This implies that at most, one unit of flow leaves the vertex:

$$\forall t, i \sum_{j \in N(i)} f_{i,j}^t \leq 1. \tag{2}$$

Third, the flows must be positive:

$$\forall t, i, j f_{i,j}^t \geq 0. \tag{3}$$

Fourth, we are looking for two paths in the graph. To enforce this, we introduce the source vertex v_{src}, which is linked to all vertices of the first time instant and the target vertex v_{trg}, which is linked to all vertices of the last time instant. The flow is from the source vertex to the target vertex. The source vertex generates exactly two units of flows and the target vertex absorbs exactly two units of flow:

$$\sum_{j \in N(src)} f_{src,j} = 2, \sum_{j \in N(trg)} f_{j,trg} = 2. \tag{4}$$

For brevity we do not explicitly write the flow conservation constraints for these vertices.

Fifth and last, the minimal distance between the frontier points can often be bounded. Consequently, we look for two paths in the graph that are C pixels far from each other. This constant can be adjusted according to the specific setting, although in practice one fixed constant is sufficient for all camera pairs over all datasets, see Section 4.2.4. For a given vertex $v_i^t = (x, x')$, let $D(i) \subset \{1...M\}$ be the nearby vertices at the same time instant:

$$D(i) = \{j | v_j^t = (y, y'), min\{d(x, y), d(x', y')\} < C\},$$

where d is the Euclidean distance and C is the constant representing the required distance between the frontier points. The last constraint is therefore:

$$\forall t, k \sum_{j \in N(k)} f_{k,j}^t + \sum_{m \in D(k)} \sum_{n \in N(m)} f_{m,n}^t \leq 1. \tag{5}$$

3.2. Linear Integer Program

We assume that we have an estimator for the probability that a given pair of critical points across views is the corresponding frontier points:

$$p_i^t = P(w_i^t = 1). \tag{6}$$

In addition, let us further assume that we can also estimate the conditional probability of a true match in a vertex given a true match in a predecessor vertex:

$$p_{i,j}^{t-1} = P(w_j^t = 1 | w_i^{t-1} = 1). \tag{7}$$

Our objective is to obtain a set of binary assignments $\mathbf{w}$ that can best explain our estimations:

$$\mathbf{w}^* = \underset{\mathbf{w}\in\mathbf{S}}{\text{argmax}}\, P(\mathbf{w}), \tag{8}$$

where $\mathbf{S}$ is the space of feasible solutions satisfying constraints (1)-(5). Assuming that our model obeys the Markov property, objective (8) can be written as:

$$\mathbf{w}^* = \underset{\mathbf{w}\in\mathbf{S}}{\text{argmax}} \sum_{t,i} logP(w_i^t) + \sum_{t,j,i\in N(j)} logP(w_j^t|w_i^{t-1}). \tag{9}$$

Because w_i^t is a binary variable, we can write

$$\begin{aligned} logP(w_i^t) &= w_i^t logP(w_i^t=1) + (1-w_i^t) logP(w_i^t=0) \\ &= w_i^t log(\frac{p_i^t}{1-p_i^t}) + J(p_i^t), \end{aligned} \tag{10}$$

where $J(p_i^t)$ is a term that does not depend on w_i^t. Similarly,

$$logP(w_j^t|w_i^{t-1}) \propto f_{i,j}^{t-1} log(\frac{p_{i,j}^{t-1}}{1-p_{i,j}^{t-1}}). \tag{11}$$

Plugging Eqs. (10)-(11) into Eq. (9), ignoring the terms that do not depend on $\mathbf{w}$, and expressing w_t^i in terms of flows, we obtain the following Integer Program:

$$\begin{aligned} \underset{\mathbf{f}}{\text{maximize}} \quad & \sum_{t,i} log(\frac{p_i^t}{1-p_i^t}) \sum_{j\in N(i)} f_{i,j}^t + \\ & \sum_{t,j,i\in N(j)} log(\frac{p_{i,j}^{t-1}}{1-p_{i,j}^{t-1}}) f_{i,j}^{t-1} \\ \text{subject to } & \forall t,i,j\ f_{i,j}^t \geq 0, f_{i,j}^t \in \{0,1\} \\ & \forall t,i \sum_{j\in N(i)} f_{i,j}^t \leq 1, \\ & \forall t,j \sum_{i\in N(j)} f_{i,j}^{t-1} - \sum_{k\in N(j)} f_{j,k}^t = 0 \\ & \sum_{j\in N(src)} f_{src,j} = 2\,, \sum_{j\in N(trg)} f_{j,trg} = 2 \\ & \forall t,k \sum_{j\in N(k)} f_{k,j}^t + \sum_{m\in D(k)} \sum_{n\in N(m)} f_{m,n}^t \leq 1 \end{aligned} \tag{12}$$

3.3. Optimization

Using standard LP solvers, a solution can be found for our Integer Program (12). Because solving IP is NP-complete, finding such an exact solution is not feasible for real-life applications. One can relax the problem into a continuous Linear Program and obtain a solution at a polynomial time, but the constraints matrix of (12) is not Totally Unimodular [24] and it is not likely to converge to the original optimal solution. However, we show that in our case, the optimal integer solution can always be computed in a much better way than the brute-force approach. This is stated in the following theorem:

Theorem 1 *The optimal integer solution of* (12) *can be recovered in* $O(TK^4)$*, where* T *is the number of time instants and* K *is the maximum number of vertices at each time instant.*

The proof is given in the supplementary material. However, such an approach can only be applied for moderately sized problems and is not scalable. Therefore, in the following we show how to compute the optimal solution in an efficient way that can also be applied to large size problems. We show that except for degenerated cases, the solution is the optimal one. It has been applied successfully for all camera pairs over all the datasets.

Similarly to [36, 9], we iteratively use the shortest path algorithm to solve the problem. This approach is also discussed in [25].

We construct a directed acyclic graph (DAG) $G' = (V', E')$ with the same structure as the graph G. An edge $e'^t_{i,j}$ is assigned the weight:

$$u(e'^t_{i,j}) = -log(\frac{p_j^{t+1}}{1-p_j^{t+1}}) - log(\frac{p_{i,j}^t}{1-p_{i,j}^t}). \tag{13}$$

The weights for the edges outgoing from the source vertex are assigned only the first term from (13), and the weights for the edges incoming to the target vertex are set to zero. The optimal solution to our Integer Program $\mathbf{f}^*$ can be written on the graph G' as:

$$\mathbf{f}^* = \underset{f\in S}{\text{argmin}} \sum_{t,i,j\in N(i)} u(e'^t_{i,j}) f_{i,j}^t,$$

where S is the set of feasible solutions of (12), satisfying the constraints given in Eqs. (1)-(5). To recover the optimal solution, we use the following:

- Find the shortest path s_1 on G'.
- For each vertex in the shortest path $v_i'^t \in s_1$, set the outgoing edges of the vertex and the nearby vertices $\{v'|v' \in D(i)\}$ to ∞.
- Find the new shortest path s_2 on the modified graph.
- Return $\hat{f} = \{s_1 \cup s_2\}$.

The above procedure is much more efficient than recovering the optimal integer solution and it can be implemented easily. In our case, we have a trellis graph and the shortest path is computed by dynamic programming [6]. Assuming there are at most K vertices at each time instant and

	Boxer	Girl	Street	Kung-Fu
# Pairs	6	28	15	300
# Frames	778	200	250	200
Type	Real	Real	Real	Graphics

Table 1. Dataset properties

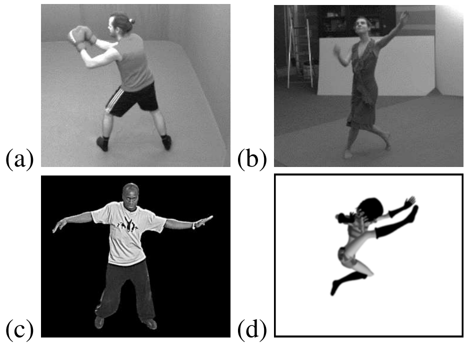

Figure 4. The datasets. (a) Boxer. (b) Girl. (c) Street (Dancer). (d) Kung-Fu.

T time instants, the solution is at a cost of $O(TK^2)$, instead of $O(TK^4)$ for the optimal integer solution. The key drawback is that it does not always guarantee that the global maximum can be recovered. However, as stated in the following theorem, unless there is a degenerate configuration, the solution will always be converged to the global one.

Theorem 2 *Let $f^* = \{s_1^* \cup s_2^*\}$ be the optimal two-path solution for the Integer Program* (12) *and let C be the constant selected for constraint* (5)*. Then, $f^* \neq \hat{f}$ if and only if (a) $s_1^* \neq s_1$ and $s_2^* \neq s_2$ and (b) $\exists t, i, j$ s.t. $v_i^t \in s_1^*, v_j^t \in s_2^*$ and $d(v_i^t, v_j^t) < 2C$.*

The proof for the above theorem is given in the supplementary material. The outcome is that the optimal solution will not be recovered only if the two pairs of frontier points at the same time instant will be less than $2C$ pixels apart. In practice, one fixed constant is sufficient for all the camera pairs over all datasets (see Section 4.2.4).

4. Experiments

We compared our method with the state-of-the-art approaches of Ben-Artzi et al.[7] and Sinha et al.[33]. For accurate comparisons, we followed the exact same procedures as in Ben-Artzi et al.[7]. The fundamental matrix F was computed by RANSAC-based sampling, based on the putative points correspondences. It is optimized using the same non-linear Levenberg-Marquardt (LM) optimization procedure.

4.1. Datasets

We used four datasets. The datasets are Boxer [5], Girl [1], Street (Dancer) [35], and Kung-Fu [2]. Table 1 presents the properties of each dataset and Fig. 4 shows sample images. All datasets used are publicly available, along with ground truth calibrations.

4.2. Implementation Details

We implemented both approaches in MATLAB using standard libraries on a computer with i7 quad-core CPU and 8GB memory. The Linear Integer Program was formulated using CVX [21] and solved with MOSEK solver [4]. We found the shortest path in our trellis by dynamic programming [6] and in all experiments, we report the results of the fast iterative algorithm.

As our input, we used two sources of information. The first is the coordinates of the critical points in each image, computed by the intersection of the convex hull of each silhouette. The second is a similarity measure of the correspondence between two critical points across views. This was used to produce $p_i^t, p_{i,j}^t$ of Eqs. (6)-(7), which will be described next.

4.2.1 The Conditional Probability Estimator

We constructed $p_{i,j}^t$, based on the distance between the two pairs. Let $v_i^t = (x, x')$ be the pair of critical points with coordinate x in one image and x' in the other image, at time instant t. Let $v_j^{t+1} = (y, y')$ be the pair of critical points at time instant $t+1$. The conditional probability estimator is according to the assumption that the Euclidean distance between the coordinates is a random variable distributed normally with zero mean and unit variance:

$$d([x, x'], [y, y']) \sim \mathcal{N}(0, 1),$$

where $\mathcal{N}(\cdot)$ is the normal distribution and $[\cdot, \cdot]$ denotes concatenations of vectors.

4.2.2 The Similarity Estimator

Assume that we have access to a similarity measure $sim(l, l')$, such that it gives us an estimate of how likely these two lines l, l' are corresponding epipolar lines. Assume that v_k^t represents the critical pair (x, x'). For v_k^t to represent a true correspondence between frontier points across views, the corresponding epipolar lines must be incident to these points. Since we do not know the exact tangent line, for each critical point there is a set of possible tangent lines. This is the set of lines $L = \{l_i\}_{i=1}^K$ from the *tangent envelope* of the silhouette at time instant t, which are incident to the critical point x. Similarly, consider L' for x'. We define the similarity estimator as:

$$p_i^t \propto max(\{sim(l, l') | l \in L, l' \in L'\}).$$

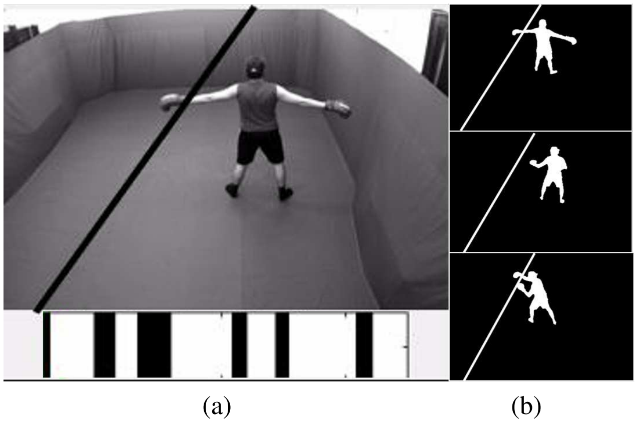

Figure 5. The Motion Barcode descriptor used for the similarity of epipolar lines. (a) The image of the boxer and a selected (arbitrary) line. Over time, at each instant the intersection of the line with the silhouette is tested and recorded as zero or one, accordingly. The recorded series of zero/one is illustrated at the bottom of the image, where black is one and white is zero. (b) Three different silhouettes at different time instants. In the first and the last time instants the motion barcode is recorded as one and for the second time instant it is recorded as zero; thus it is $[1, 0, 1]$.

4.2.3 Motion-Based Similarity of Epipolar Lines

Here we briefly describe the similarity measure for two lines to represent the corresponding epipolar lines. It is the input to our similarity estimator. It was used in [7, 26, 8], which also provides a detailed description.

The similarity measure is based on a descriptor denoted as *motion barcode*. For a given image line, a motion barcode is constructed as follows. For each image in the sequence, the intersection of the line with the silhouette is tested. The value of the motion barcode at that time instant is set to either zero or one, accordingly. Thus, it is a binary sequence of the same length as the number of images in the sequence. It has been shown that if two lines are indeed corresponding epipolar lines, their motion barcodes should be very similar [7]. The similarity between the motion barcodes of two lines is the correlation between the binary sequences. The motion barcode is illustrated in Fig. 5.

4.2.4 The Graph

We set the required minimal distance between the frontier points C for $D(\cdot)$ to 15 pixels, in all the experiments across all datasets and camera pairs, without fine-tuning it for each camera pair individually. The required distance depends on the specific setting and can in principle be adjusted for each specific case accordingly. We verified that the same single constant is valid for all valid frontier points over all the frames in the datasets.

Our method results in two paths in the graph. Each path consists of a set of vertices representing a match between frontier points across views. The flow conservation constraints required selecting vertices at each frame, even if the similarity estimator and the conditional probability estimator output have very low probabilities. Similarly to [7], we used a threshold of 0.95 on the motion barcodes similarity measure to remove unreliable matches whose associated similarity is lower than this threshold.

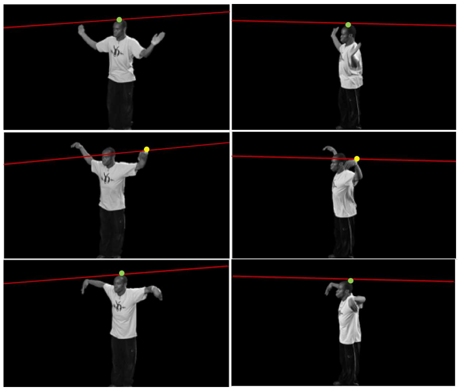

Figure 6. True correspondences of non-frontier points. On rare occasions, less than 1%, our method also matches non-frontier points. Each row is two views from the same time instant, ordered by time from the top downward. The red lines are the epipolar lines. The first and last rows from the top show matched frontier points, denoted in green. The middle row is the time instant with non-frontier points that are matched, denoted in yellow. These are true corresponding critical points. See text for more details.

In addition to recovering frontier points, on rare occasions our method also matches non-frontier points. Fig. 6 shows a time instant for which non-frontier points are matched. It presents samples from eight consecutive frames, ordered by time from the top downward; each row shows corresponding views for the same time instant. The red lines denote the corresponding epipolar lines. The actor raises his hands and then lowers them. At the time instances represented by the first and last rows from the top, the true frontier point correspondences are recovered, denoted by green circles. At the time instant represented by the second row from the top, the matched points denoted by yellow circles are indeed true corresponding points but they are not frontier points. These cases are less than 1% of the matches and occur when a) the similarity estimator outputs for these critical points which are non-frontier points very high probability to be a true correspondence and b) the corresponding true frontier points are matched in nearby positions, immediately before and after this specific time instant. Due to the regularization, the non-frontier points are preferred over the frontier points.

The smoothness constraint on the coordinates of the frontier point across frames may not always hold, mainly in cases where an abrupt motion is being carried. Considering such a motion as a distance of more than 30 pixels between consecutive corresponding frontier points, it is then occurs in approximately 17% of the cases. In such cases the performance of our approach depends on the ability of the motion-barcode feature to clearly distinguish between true and false correspondences.

4.3. Evaluation

4.3.1 Metrics

Quality of Fundamental Matrix. A standard metric for assessing the quality of the fundamental matrix is the symmetric epipolar error [22] defined as:

$$Q(F) = \frac{1}{N}\sum_{i} d(x'_i, Fx_i)^2 + d(x_i, F^T x'_i)^2,$$

where F is the evaluated fundamental matrix and $\{(x_i, x'_i)\}_{i=1}^{N}$ are the corresponding points across views.

Efficiency. The efficiency measure is the expected number of RANSAC iterations needed to achieve a given or a better quality of a fundamental matrix (a lower error).

4.3.2 Methodology

Efficiency Comparison. For the generation of the efficiency metric, we followed the protocol by [7]. They selected the best error every 1000 iterations and optimized it using non-linear Levenberg-Marquardt (LM) optimization procedure. For example, in the Girl dataset there are 28 camera pairs and therefore, our sample size is 2800 errors. We then calculated the cumulative distribution function (CDF) of the error and used it in the evaluation as in the baselines.

Inliers' Probabilities. The efficiency metric used by [7] is based on (a) the probability of having an inlier, (b) the selection of the best sample using the ground truth points, and (c) the non-linear optimization technique. Thus, it might not directly reflect the quality of the putative correspondences using the tested approach. We therefore present the probability of having an inlier for various thresholds using our approach, which can be used for future reference. It is calculated as follows. We measured the symmetric epipolar distance for each recovered corresponding pair of points using our approach with respect to the ground truth fundamental matrices. Using this distance, the fraction of inliers can be evaluated for each dataset and required accuracy.

Epipolar Distance		1	0.8	0.5	0.4	0.3	0.2
Boxer	Sinha	2.9M	2.9M	-	-	-	-
	Ben-Artzi	5K	12K	111K	996K	-	-
	Ours	1230	1354	13K	300K	600K	-
Girl	Sinha	149K	388K	13M	-	-	-
	Ben-Artzi	4K	9K	129K	918K	13.7M	-
	Ours	828	867	1195	4562	30283	560K
Street	Sinha	159K	340K	1.8M	7.4M	-	-
	Ben-Artzi	7K	20K	255K	616K	1.2M	-
	Ours	928	959	1279	2142	6245	62.5K
Kung-Fu	Sinha	65K	134K	822K	1.9M	8.6M	-
	Ben-Artzi	2K	4K	23K	71K	302K	-
	Ours	711	720	814	998	2058	13.4K

Table 2. The expected number of RANSAC iterations required to reach a given accuracy of the fundamental matrix, using our approach and the baselines. In each dataset, the number of iterations is averaged over all cameras pairs. Accuracy is measured using the symmetric epipolar distance with respect to the ground-truth points. The best hypothesis is selected every 1000 RANSAC iterations and is optimized using non-linear Levenberg-Marquardt (LM) method. Empty cells indicate that the required accuracy was not attained.

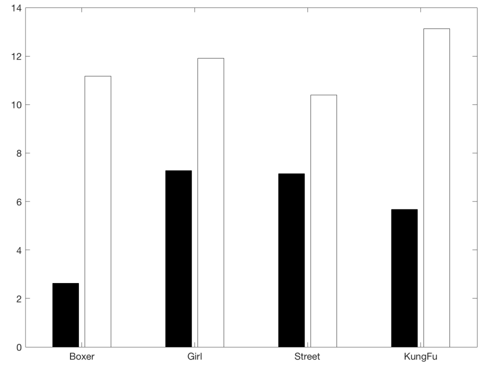

Figure 7. Ratio of the average running time between our method and the baselines. The x-axis is the datasets and the y-axis is the $log2$ running time ratio. White cells represent Sinha's method and black cells represent Ben-Artzi's method. Time is averaged over all cameras pairs and over all accuracy levels presented in Table 2. The runtime was measured on Intel Xeon Server E5-2650 2.3GH, implemented in MATLAB.

Overhead. Our method requires an additional step of finding the two paths on the graph. The runtime cost is equivalent to $300 - 1200$ RANSAC iterations, depending on the length of the sequence and the number of vertices. It was added to the comparisons.

4.4. Results

Table 2 presents the expected number of RANSAC iterations per accuracy using our method and the baselines. Following [7], the best hypothesis is selected every 1000 RANSAC iterations and is optimized using the non-linear (LM) method. The accuracy is the symmetric epipolar distance with respect to ground-truth points. A very high accuracy can be reached very quickly. For example, reaching an accuracy of 0.3 for the Street dataset requires on average

Epipolar Distance	1	0.8	0.5	0.4	0.3	0.2
Boxer	0.37	0.30	0.19	0.15	0.11	0.07
Girl	0.46	0.38	0.24	0.19	0.15	0.11
Street	0.66	0.53	0.35	0.28	0.21	0.14
Kung-Fu	0.65	0.56	0.39	0.32	0.25	0.17

Table 3. The inliers probabilities for each required accuracy and for each dataset, averaged over all camera pairs in the dataset. It was measured by determining the symmetric epipolar distance of our recovered points with respect to the ground truth fundamental matrices.

Epipolar Distance	1	0.8	0.5	0.4	0.3	0.2
Mean Inliers prob.	0.53	0.44	0.29	0.23	0.18	0.12
Required Samples	390	1440	26K	135K	752K	1.28M

Table 4. First row. The mean probability of having an inlier of a given accuracy for all camera pairs over all datasets. Second row. The required number of samples using the RANSAC procedure for reaching the required accuracy. On average, only 26K samples are needed to reach a sub-pixel accuracy of 0.5.

Accuracy \Dataset	Boxer	Girl	Street	Kung-Fu
Any	547	108	127	150
0.5	104	26	44	58
0.3	60	16	27	37

Table 5. The number of corresponding points recovered for each dataset, over all the camera pairs. The first row is the total number of points recovered. The second row is the number of inliers having a symmetric epipolar distance equal or less than 0.5, and the third row is for a symmetric epipolar distance of 0.3.

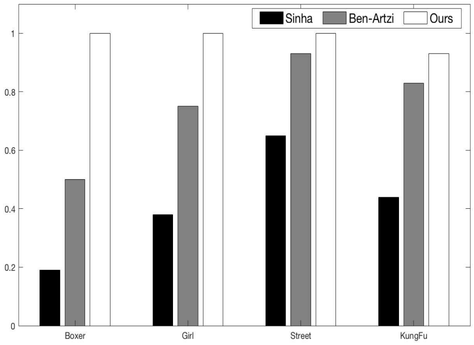

Figure 8. The fraction of camera pairs whose fundamental matrices reached a required symmetric epipolar distance of 0.8. The x-axis is the dataset and the y-axis is the fraction of the camera pairs that reached the required accuracy. Using our approach, all camera pairs, in all datasets except Kung-Fu converged. For the Kung-Fu dataset, 88% of the cameras converged. See text for more details.

only 6245 RANSAC iterations. Generally, the higher the required accuracy, the more the improvement can be introduced. Our approach is the only one to reach a very high accuracy level of 0.2.

Fig. 7 presents the ratio between the average runtime of our method and the baselines. For each dataset and for each method, time is averaged over all camera pairs and accuracy levels presented in Table 2. Our approach introduces an improvement of orders of magnitude, for each datasets and for all accuracy levels. Overall, the mean improvement is 92.82 with respect to [7] and 993.71 with respect to [33].

Table 3 presents the inliers' probabilities. It was computed based on the symmetric epipolar distance to the ground truths fundamental matrices. The probability for each required accuracy is the average over all camera pairs in the dataset. Table 4 shows the mean probability of having an inlier of a required accuracy for all camera pairs over all datasets. It also shows the required number of RANSAC samples needed to reach each accuracy. On average, only 1440 samples are needed to reach a sub-pixel accuracy of 0.8. For a required accuracy of 0.5, only $26K$ samples are needed on average. Table 5 shows the number of recovered putative corresponding pairs for a given accuracy using our approach, based on the inlier probabilities. For the boxer dataset, we were able to recover more than 100 pairs with an accuracy of 0.5 and 60 pairs with an accuracy of 0.3.

Fig. 8 shows the percentage of converged cameras using our method and the baselines, for a required accuracy of 0.8. In all datasets except Kung-Fu, using our approach, all camera pairs converged. In the Kung-Fu dataset, however, 88% of the camera pairs converged. This is due to the fact that there are camera pairs for which the epipoles are inside the convex hull. In such cases, tangent-based methods often fail to recover accurate matching points. These cases of failure are inherent in such approaches, see [7] for an illustrative example. Nevertheless, due to the dynamic nature of the object there are often frames within the same sequence where the epipoles are also outside the convex hull, which is sufficient for calibration. For example, for a required accuracy of 1.5, only two camera pairs failed in the Kung-Fu dataset and all other camera pairs over all the other datasets converged.

5. Conclusion

We introduced a graphical model for calibrating a multi-camera system from the motion of silhouettes. Our approach recovers corresponding points efficiently and accurately, outperforming state-of-the-art methods by several orders of magnitude. It is optimized very efficiently, providing a practical solution. Our approach fits seamlessly into a silhouettes-based pipeline, and it can be used automatically each time a new sequence of silhouettes is captured.

Acknowledgement. The author would like to thank Ronen Basri for helpful comments on an earlier version of this manuscript.

References

[1] Dancer dataset, inria, 4d-repository , http://4drepository.inrialpes.fr/public/viewgroup/1, 2010. 5

[2] Data set, kung-fu girl, http://www.mpi-inf.mpg.de/ departments/irg3/kungfu/, 2010. 5

[3] E. Aganj, J.-P. Pons, F. Ségonne, and R. Keriven. Spatio-temporal shape from silhouette using four-dimensional delaunay meshing. In *ICCV'07*, pages 1–8, 2007. 2

[4] M. ApS. *The MOSEK optimization toolbox for MATLAB manual. Version 7.1 (Revision 28).*, 2015. 5

[5] L. Ballan and G. M. Cortelazzo. Multimodal 3d shape recovery from texture, silhouette and shadow information. In *3D Data Processing, Visualization, and Transmission, Third International Symposium on*, pages 924–930. IEEE, 2006. 5

[6] R. Bellman. On the theory of dynamic programming. *Proceedings of the National Academy of Sciences*, 38(8):716–719, 1952. 4, 5

[7] G. Ben-Artzi, Y. Kasten, S. Peleg, and M. Werman. Camera calibration from dynamic silhouettes using motion barcodes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 1, 2, 5, 6, 7, 8

[8] G. Ben-Artzi, M. Werman, and S. Peleg. Event retrieval using motion barcodes. In *ICIP'15*. 2, 6

[9] R. Bhandari. *Survivable networks: algorithms for diverse routing*. Springer Science & Business Media, 1999. 4

[10] E. Boyer. On using silhouettes for camera calibration. In *ACCV'06*, pages 1–10. 2006. 2

[11] K. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. In *CVPR'03*, volume 1, pages I–77, 2003. 2

[12] R. Cipolla, K. E. Astrom, and P. J. Giblin. Motion from the frontier of curved surfaces. In *ICCV'95*, pages 269–275, 1995. 1, 2

[13] R. Cipolla and P. Giblin. *Visual motion of curves and surfaces*. Cambridge University Press, 2000. 2

[14] M.-A. Drouin, P.-M. Jodoin, and J. Prémont. Camera-projector matching using an unstructured video stream. In *CVPR'10 Workshop*, pages 33–40. IEEE, 2010. 2

[15] E. B. Ermis, P. Clarot, P.-M. Jodoin, and V. Saligrama. Activity based matching in distributed camera networks. *IEEE Trans. IP*, 19(10):2595–2613, 2010. 2

[16] M. A. Fischler and R. C. 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. 1

[17] K. Forbes, F. Nicolls, G. De Jager, and A. Voigt. Shape-from-silhouette with two mirrors and an uncalibrated camera. In *ECCV'06*, pages 165–178. 2006. 2

[18] J.-S. Franco and E. Boyer. Exact polyhedral visual hulls. In *BMVC'03*, volume 1, pages 329–338, 2003. 2

[19] Y. Furukawa, A. Sethi, J. Ponce, and D. Kriegman. Structure and motion from images of smooth textureless objects. In *European Conference on Computer Vision*, pages 287–298. Springer, 2004. 1

[20] Y. Furukawa, A. Sethi, J. Ponce, and D. Kriegman. Robust structure and motion from outlines of smooth curved surfaces. *IEEE Trans. PAMI*, 28(2):302–315, 2006. 2

[21] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`, Mar. 2014. 5

[22] R. Hartley and A. Zisserman. Multiple view geometry in computer vision. 2003. 7

[23] C. Hernández, F. Schmitt, and R. Cipolla. Silhouette coherence for camera calibration under circular motion. *IEEE Trans. PAMI*, 29(2):343–349, 2007. 2

[24] A. J. Hoffman and J. B. Kruskal. Integral boundary points of convex polyhedra. In *50 Years of Integer Programming 1958-2008*, pages 49–76. Springer, 2010. 4

[25] F. Iqbal and F. A. Kuipers. Disjoint paths in networks. *Wiley Encyclopedia of Electrical and Electronics Engineering*. 4

[26] Y. Kasten, G. Ben-Artzi, S. Peleg, and M. Werman. *Fundamental Matrices from Moving Objects Using Line Motion Barcodes*, pages 220–228. Springer International Publishing, Cham, 2016. 2, 6

[27] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Trans. PAMI*, 16(2):150–162, 1994. 2

[28] S. Lazebnik, Y. Furukawa, and J. Ponce. Projective visual hulls. *International Journal of Computer Vision*, 74(2):137–165, 2007. 1

[29] P. R. S. Mendonca, K.-Y. Wong, and R. Cippolla. Epipolar geometry from profiles under circular motion. *IEEE Trans. PAMI*, 23(6):604–616, 2001. 2

[30] G. Miller and A. Hilton. Exact view-dependent visual-hulls. In *ICPR'06*, pages 107–111, 2006. 2

[31] D. Pundik and Y. Moses. Video synchronization using temporal signals from epipolar lines. In *European Conference on Computer Vision*, pages 15–28. Springer, 2010. 2

[32] P. Ramanathan, E. G. Steinbach, and B. Girod. Silhouette-based multiple-view camera calibration. In *VMV*, pages 3–10, 2000. 2

[33] S. N. Sinha and M. Pollefeys. Camera network calibration and synchronization from silhouettes in archived video. *IJCV*, 87(3):266–283, 2010. 1, 2, 5, 8

[34] S. N. Sinha, M. Pollefeys, and L. McMillan. Camera network calibration from dynamic silhouettes. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE. 1

[35] J. Starck and A. Hilton. Surface capture for performance-based animation. *Computer Graphics and Applications*, 27(3):21–31, 2007. 5

[36] J. Suurballe. Disjoint paths in a network. *Networks*, 4(2):125–145, 1974. 4

[37] K.-Y. Wong and R. Cipolla. Structure and motion from silhouettes. In *ICCV'01*, volume 2, pages 217–222, 2001. 2

[38] H. Yamazoe, A. Utsumi, and S. Abe. Multiple camera calibration with bundled optimization using silhouette geometry constraints. In *ICPR'06*, volume 3, pages 960–963. IEEE, 2006. 2

[39] H. Zhang and K.-Y. Wong. Self-calibration of turntable sequences from silhouettes. *IEEE Trans. PAMI*, 31(1):5–14, 2009. 2