Direct-from-Video: Unsupervised NRSfM

Karel Lebeda(✉), Simon Hadfield, and Richard Bowden

Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford, UK
{k.lebeda,s.hadfield,r.bowden}@surrey.ac.uk

Abstract. In this work we describe a novel approach to online dense non-rigid structure from motion. The problem is reformulated, incorporating ideas from visual object tracking, to provide a more general and unified technique, with feedback between the reconstruction and point-tracking algorithms. The resulting algorithm overcomes the limitations of many conventional techniques, such as the need for a reference image/template or precomputed trajectories. The technique can also be applied in traditionally challenging scenarios, such as modelling objects with strong self-occlusions or from an extreme range of viewpoints. The proposed algorithm needs no offline pre-learning and does not assume the modelled object stays rigid at the beginning of the video sequence. Our experiments show that in traditional scenarios, the proposed method can achieve better accuracy than the current state of the art while using less supervision. Additionally we perform reconstructions in challenging new scenarios where state-of-the-art approaches break down and where our method improves performance by up to an order of magnitude.

Keywords: Non-rigid SfM · Structure from motion · Visual tracking · Template-free · Gaussian process

1 Introduction

Non-Rigid Structure-from-Motion (NRSfM) is a problem which has attracted considerable interest in recent years, from application areas such as medical imaging and the special effects industry. The problem is usually formulated as the estimation of camera motion and of a time-varying 3D shape for an *a priori* unknown object, using only a set of 2D point trajectories [1–4]. We propose a modified formulation, where the task is completely unsupervised (with the only input being a selection of what object to model). In other words, our task is to estimate the camera motion and time-varying 3D shape of an *a priori* unknown object from a previously unseen video-sequence, using only a bounding box in the first frame. As far as the authors are aware, there is no previous work addressing simultaneous tracking and non-rigid modelling from a monocular camera.

Electronic supplementary material The online version of this chapter (doi:10.1007/978-3-319-49409-8_50) contains supplementary material, which is available to authorized users.

G. Hua and H. Jégou (Eds.): ECCV 2016 Workshops, Part III, LNCS 9915, pp. 578–594, 2016.
DOI: 10.1007/978-3-319-49409-8_50

The NRSfM problem is very challenging, due to the ambiguous separation of 2D observations into rigid camera motion and non-rigid object deformation. This is exacerbated in the unsupervised scenario, where the observations are noisy, contain outliers (due to matching failure) and may even belong to background clutter. Despite these issues, we are able to successfully address the problem by adapting techniques from 3D visual tracking. Online estimates of camera trajectory and object shape can be fed back, to improve the accuracy of the point tracking as the sequence progresses.

Another major contribution of this paper is that the traditional 3D object model (defined as a 3D point cloud) is upgraded to a continuous 3D surface using Gaussian Process shape modelling [5]. This makes it possible to segment the object from the background, to reason about self-occlusions, and to intelligently sample points in regions of low confidence (due to the probabilistic nature of the model).

To the best of our knowledge, all state-of-the-art NRSfM techniques use 2D point tracks as their input (with varying levels of density). In this publication, we present a unified framework which jointly addresses the problems of 2D point tracking and NRSfM directly on video frames. The additional 3D information improves the 2D tracking far beyond what is possible from a generic stand-alone system. In turn these more accurate point tracks help to refine future NRSfM estimates.

The 2D tracks required by state-of-the-art techniques are often precomputed (or taken from known annotations). For this precomputation, it is common to work with a reference template or video frame, against which all other frames are registered. This is important as the concatenation of frame-to-frame correspondences (*e.g.* from optical flow) inevitably leads to an accumulation of errors (drift). However, reference frames limit the possible applications of the technique. In contrast, we address the problem of track drift explicitly, using multiple overlapping (both spatially and temporally) sets of dense trajectories, in addition to easily localised sparse trajectories for long-term consistency. This obviates the need for a reference frame, and makes it possible to process a wider range of scenarios. These include strong rotations and self-occlusions, where there may be zero overlap between the first frame and some frames later in the video.

Even though the proposed technique does not require any supervision (beyond a single target bounding box), it extends easily to the more traditional supervised scenarios using precomputed 2D tracks. Additional point correspondences (such as tracks of SIFT features, regressed facial landmarks, *etc.*) can be exploited within the framework to further improve performance.

One major issue in NRSfM research is the regularisation of non-rigid object deformations. With unconstrained deformation, there is a trivial solution for any set of observations, where the camera does not move and observations are explained by complicated object deformations. To prevent this, the shape deformation is usually defined as arising from a weighted combination of basis shapes. In this paper we employ a novel set of constraints and regularisation, which ensure that every basis shape represents an extreme (but feasible) pose of the target object. Shape deformations are then constrained to lie within the *feasible* manifold (a convex subspace) formed by these basis shapes. This regularisation renders the method very robust to overfitting.

To summarise, the primary contributions of this paper are: (1) a unified framework to jointly solve the online (although not real-time), direct, template-free NRSfM and point tracking tasks, (2) the use of Gaussian Process shape model and (3) novel constraints to regularise the basis shape selection. The source code of the method will be made available online.

2 Related Work

Most approaches to NRSfM are factorisation-based [6], as introduced by Bregler et al. [7]. To simplify the problem, the orthographic camera model is used [3,4,8,9]. This way, the 2D point locations (per frame) can be expressed as an affine function of the 3D locations, which are in turn a linear combination of *basis shapes*. The set of projection equations (for each 3D point and video frame where it is visible) is then rewritten as a matrix-matrix multiplication. The projection multiplication is decomposed (usually using SVD) back to the factors, yielding the camera parameters (translation and rotation, for each frame), basis shape mixing parameters (i.e. coefficients of the linear combination, for each frame) and basis shape locations (for each point).

This problem is inherently ill-posed, having significantly more unknowns than equations. To render it solvable, additional constraints are applied. In the original paper [7], the low-rank constraint was applied, effectively setting/limiting the number of basis shapes. All following approaches use this constraint and apply additional constraints, priors, heuristics and regularisations. These include spatial smoothness of shape [3,10–12] (the points lying close to each other in 2D tend to lie close to each other in 3D); temporal smoothness of shape [1,3,9,10] (the shape changes smoothly over time); temporal smoothness of camera poses [1,3] (the camera trajectory is smooth in time); and inextensibility [11,12] and other physics-based priors [1,13]. In this paper we propose an additional constraint, that each basis shape must relate to a feasible target pose, greatly improving the stability of the optimisation.

One limitation of the factorisation-based formulation is that it is conditional on all 2D tracks spanning the length of the video. This condition is removed by either estimating the missing data [14] or using methods based on Bundle Adjustment (BA) [1,10], such as the proposed method. In this case, matrix factorisation is replaced with global optimisation of the model parameters (basis shapes, mixing coefficients and camera trajectory). Another reason for the use of Bundle Adjustment is its ability to solve for more complicated camera models. Finally, BA-based techniques also scale well in terms of memory and computation time.

Table 1 compares the properties of selected state-of-the-art NRSfM approaches. Although there are many more works, this comparison captures general trends which can be observed in the field. All current techniques use either a template, a precomputed set of 2D trajectories, or an RGBD camera to address the task. To the best of the authors' knowledge, there has been no prior

Table 1. Comparison of state-of-the-art approaches for dynamic shapes reconstruction.

Property	Zollhofer [15]	Newcombe [16]	Garg [17]	Yu [18]	Proposed
Template-free		✓	✓		✓
Direct	✓	✓		✓	✓
Monocular RGB			✓	✓	✓
Online	✓	✓		✓	✓

approach to solve NRS*f*M which would be at the same time direct, template-free and using only a single RGB camera.

3 Method

In this section, we present our novel formulation of the NRS*f*M problem. See Fig. 2 for an overview of the proposed algorithm. Its input is a video-sequence and optionally additional (independently estimated) trajectories. Its outputs are the camera trajectory, reconstructed basis shapes (point clouds) and the mixing parameters for each frame. From these, the time-varying shape can be reconstructed at any frame (*i.e.* the *instantaneous shape*). Optionally, the shape can be extracted in the form of an explicit polygonal mesh, parametrisable by the coefficients.

As the first step (line 1 in Fig. 2) for the first frame, a bounding box is used to specify the target. Within this boundary, sparse and dense 2D features are extracted (lines 2 and 3) as detailed in Sect. 3.1. Optionally, further supervision points can be supplied (line 4) from another source (such as regressed landmarks in the case of a face sequence). These 2D points are backprojected to the dense object model (see Sect. 3.3 for details) and then duplicated K times to form the initial basis shapes (line 5). The mixing coefficients are initialised to 1/K. For more details on how the (time-varying) point clouds are represented, see Sect. 3.2.

On every subsequent frame, we first track the existing 2D features in the new image frame, as specified in Sect. 3.1 (lines 7 and 8). Using these 2D tracks and their 3D correspondences, we estimate the current camera parameters (line 10). Unless the camera has undergone significant motion (line 11 in Fig. 2), the algorithm continues processing the next frame.

In the case where the camera has moved far enough since the last Bundle Adjustment to provide a sufficient baseline for depth estimation, we jointly optimise (line 13) all the variables in the system: basis shapes, their per-frame mixing coefficients up to the current time and the camera trajectory to the current frame. Bundle Adjustment is preferred over filtering and other methods since it provides better performance given the same inputs [19]. Due to the novel regularisation, the obtained basis shapes are well constrained and extremely stable, which helps avoid difficulties with the *basis ambiguity issue* [8].

After the 3D point clouds have been optimised, the implicit model is retrained (line 14). This model then provides the object/background segmentation, needed for creation of new points to be tracked. New dense tracks are initialised in the whole image region containing the target object (line 16), while new sparse tracks are initialised only where low confidence in the 3D shape renders them beneficial (line 15). This directed sampling is the main advantage of tackling tracking and reconstruction simultaneously. The corresponding 3D point clouds are initialised by back-projecting the points locations to the model. See Sects. 3.1 and 3.3 for details.

3.1 Obtaining Point Trajectories

Estimation of dense point tracks within a video sequence is inherently burdened by the drift problem: concatenation of frame-to-frame point correspondences leads to error accumulation, rendering long term dense trajectories unreliable. This is traditionally countered by having a reference frame, to which all other frames are registered, instead of concatenation. While this removes drift caused by accumulation of errors, it adds the requirement to have a single frame which overlaps all other frames. This in turn prevents application to sequences with strong rotation and self-occlusion. We instead address this problem directly, by limiting the temporal span of dense tracks to a fixed number of frames. Multiple sets of these tracks are then created, overlapping in both time and space. These are combined during the optimisation, being reconstructed in the common 3D world. Additionally, for long-term consistency, sparse features are used, which are easily localisable and can be tracked frame-to-frame more robustly. Furthermore, the visibility of these sparse points is maintained based on the dense model (*i.e.* due to self-occlusion) and the points may be redetected when they become visible again, facilitating *loop closures*.

The dense features $\mathcal{D}^t$ are sampled on a regular grid within the initial bounding box (in the first frame), or within the area of the estimated object boundary found by projecting the model into the current frame $P(\mathbf{M}|\mathbf{C}^t)$. The density of these points is set by the user to control the trade-off between processing time and level of model detail. After each BA, new dense features are created, spanning the entire area of the projected model, to ensure overlap between the subsets of dense trajectories within $\mathcal{D}^t$. The dense frame-to-frame tracks are obtained by registering feature images obtained through deep-learning [20].

For the sparse tracks $\mathcal{S}^t$, we extract SIFT and Hessian-Affine feature points, which are specifically chosen to be robustly localisable over long timescales. These are then tracked using pyramidal Lucas-Kanade. Unlike the dense features, the temporal span of the sparse tracks is unlimited. This means that we do not need to ensure spatial overlap between consecutive "batches" of tracks, as for the dense points. Indeed, it is counterproductive to sample too many sparse points within any particular region of the target object, as this results in wasted computation. To prevent this, we employ the probabilistic nature of our model which is based on Gaussian Processes and extract new features only in areas with high uncertainty of the shape (*i.e.* where the new features will be the most beneficial; see Sect. 3.3 for details).

For both sparse and dense tracks, background features may become included in either the initial bounding box or later segmentation. For this reason, feature filtering takes place, based on their reconstructed 3D location relative to the model. Features inconsistent with the model are considered outliers and are not used in further computations.

3.2 Non-rigid 3D Reconstruction

Along with the majority of state-of-the-art approaches, we express the instantaneous 3D shape $\mathtt{B}^t$ as a linear combination of basis shapes $\mathtt{B}$:

$$\mathtt{B}^t = \mathtt{B}\boldsymbol{\alpha}^t. \tag{1}$$

This instantaneous shape can be projected to find the equivalent 2D observations:

$$\hat{\mathbf{u}}^t = P\left(\mathtt{B}^t|\mathbf{C}^t\right), \tag{2}$$

i.e. every 3D point in $\mathtt{B}^t$ is projected by a camera with parameters $\mathbf{C}^t$ to create the concatenated 2D point matrix $\hat{\mathbf{u}}^t$. The camera model used in our experiments is full projective, however the approach generalises to any other camera model (*e.g.* orthographic, spherical, *etc.*) as long as it provides a unique back-projection (a 2D point to a 3D ray) for any 2D image location. This way we separate (for every frame) the rigid motion as the camera motion (captured by $\mathbf{C}^t$) and the non-rigid motion as the shape deformation (captured by $\boldsymbol{\alpha}^t$).

The common $3K$-rank constraint used extensively throughout the NRSfM literature, is equivalent to fixing the number of basis shapes to K. In this paper we introduce a novel regularisation which forces the basis shapes to be meaningful modes, or linearly independent "extremes", of the target's shape. This is done via the following constraints:

$$\mathbf{1}_K^\top \boldsymbol{\alpha} = 1 \quad \text{and} \quad \alpha_j \in [0;1], \tag{3}$$

where α_j is the j-th element of $\boldsymbol{\alpha}$. This effectively limits the targets shape to a *convex* combination of the basis shapes (*i.e.* a finite $K-1$ dimensional manifold in the full shape space, *e.g.* a triangle on a 2D hyperplane for $K = 3$). This is important during the optimisation process (see below) and is also useful for modelling and visualisation.

The projection equation provides a simple geometric error to be minimised during the rigid camera pose estimation:

$$\mathbf{C}^t = \arg\min_{\mathbf{C}} \left\|\rho\left(\mathbf{u}^t - P\left(\mathtt{B}^t|\mathbf{C}\right)\right)\right\| \tag{4}$$

where $\mathbf{u}^t$ comprises the 2D sparse, dense and supervision points, and ρ is an element-wise robust cost function, to provide outlier tolerance (similar to [21]). This is minimised using the conditional gradient method.

There are two ways in which the instantaneous 3D shape for each frame could be estimated. Firstly, the unknown set of coefficients $\boldsymbol{\alpha}^t$ could be included as parameters to Eq. (4) and estimated for each frame, jointly with the camera pose. The second approach is to postpone the estimation of the mixing coefficients

($\boldsymbol{\alpha}^t \leftarrow \boldsymbol{\alpha}^{t-1}$) until the next bundle adjustment. Empirically we find that the latter approach is more stable as it allows more observations and additional regularisation to be used to constrain the non-rigid deformations.

Theoretically, there is nothing preventing BA from being executed on every frame, however that would be excessively time-demanding (BA is the most time-consuming stage of the algorithm even with sparse execution, see Table 3). Requiring a baseline of sufficient width (non-negligible camera motion) between two consecutive BA runs creates well-timed on-request executions on keyframes characterised by equidistant camera poses.

The cost function optimised in BA is similar to (4), with several major differences:

$$\begin{gathered}\min_{\mathtt{B},\mathcal{C}^t,\mathcal{A}^t} \sum_{\tau=1}^{t} \left\|\rho\left(\mathbf{u}^\tau - P\left(\mathtt{B}\boldsymbol{\alpha}^\tau|\mathbf{C}^\tau\right)\right)\right\| + \Lambda_\alpha(\mathcal{A}^t) + \Lambda_{\mathtt{B}}(\mathtt{B}) + \Lambda_{\mathbf{C}}(\mathcal{C}^t) \\ \text{s.t.} \quad \mathbf{1}_K^\top \boldsymbol{\alpha}^\tau = 1 \quad \text{and} \quad \alpha_j^\tau \in [0;1] \quad \forall j, \tau,\end{gathered} \tag{5}$$

where $\mathcal{A}^t$ includes all mixing vectors up to frame t and $\mathcal{C}^t$ contains all cameras up to frame t. Since it is vital to update the mixing coefficients $\boldsymbol{\alpha}$ during BA, the combination of basis shapes needs to be expressed explicitly. The projection errors are summed across all the frames seen thus far (a windowed version, limited to a recent history may be considered if speed is an issue). The robust cost function ρ employed here is the Cauchy loss, as provided by the Ceres Solver [22]. Finally, there are additional priors and regularisations employed. Significant effort is given to these throughout the literature, and sometimes they constitute the major novelty of an article [10,23].

We employ the *temporal smoothness of shape* prior. This means the shape cannot change suddenly over time. This is achieved by penalising fast changes in the mixing coefficients:

$$\Lambda_\alpha(\mathcal{A}^t) = w_\alpha \sum_{\tau=2}^{t} ||\boldsymbol{\alpha}^{\tau-1} - \boldsymbol{\alpha}^\tau||^2 \tag{6}$$

where w_α is an appropriate weighting.

In the proposed method, we want the basis shapes to be extremes (rare, but feasible instances) of the shape variation. In other words, the instantaneous shapes are required to span a (convex) subspace, tightly bounded by the basis shapes. This renders the method very robust to overfitting. The first requirement, that the instantaneous shapes span a limited space, is achieved by limiting the $\boldsymbol{\alpha}$ coefficients (Eq. (3)). The second requirement, that the bounding subspace is tight around the observed poses, stems from the need to decouple the rigid and non-rigid motions. Therefore we introduce the final regularisation term:

$$\Lambda_{\mathtt{B}}(\mathtt{B}) = w_{\mathtt{B}} \sum_{i=2}^{K} \sum_{j=1}^{i-1} \|\mathtt{B}_i - \mathtt{B}_j\|^2 \tag{7}$$

where $w_{\mathtt{B}}$ is an appropriate weighting.

Finally, to enforce the prior of *temporal smoothness of camera trajectory*, a different cost is chosen. It is desirable to penalise sudden changes in camera parameters without creating an energy inhibiting free camera motion in the world. Therefore the following is used:

$$\Lambda_{\mathbf{C}}(\mathcal{C}^t) = w_{\mathbf{C}} \sum_{\tau=2}^{t} \begin{cases} 1 & \text{if} ||\mathbf{C}^{\tau-1} - \mathbf{C}^{\tau}|| \geq \theta_{\mathbf{C}} \\ 0 & \text{if} ||\mathbf{C}^{\tau-1} - \mathbf{C}^{\tau}|| < \theta_{\mathbf{C}} \end{cases}, \tag{8}$$

where $\theta_{\mathbf{C}}$ is a chosen threshold and $w_{\mathbf{C}}$ is a large (relative to the other costs) constant.

3.3 Object-Background Segmentation

For successful 2D tracking in the presence of background clutter, it is necessary to segment the object of interest from the background. The reconstructed 3D points can give a rough idea where the object is located, however they are not sufficient for segmentation. For this reason, we keep a dense model of the object. It is modelled as a Gaussian Process (GP) in polar coordinates, similarly to [24], where the distance of the surface from the object centre is a function of its bearing angles (azimuth and elevation).

The Gaussian Process is trained on the reconstructed 3D points, and is retrained after every bundle adjustment as follows. Firstly, the 3D points in a *canonical shape* (combined from the basis shapes $\mathtt{B}$ using $\boldsymbol{\alpha}'$ averaged over the history thus far) are expressed as vectors in polar coordinates, *i.e.* as a radius r and a pair of angles (θ, ϕ) per point. All the points are then used as training data, regressing the radius from the angles: $r = \text{GP}(\theta, \phi|\kappa)$, where κ is the *kernel* of the GP (in this work we use a combination of exponential, white-noise and bias kernel; its parameters are optimised to maximise the observation likelihood on the training data).

The model can be queried in any direction (θ, ϕ), yielding the local radius. As a result, the model densifies the point cloud, fitting a continuous surface to the sparse points. This way, it tells us where the object is; both in the 3D space (reasoning about occupancy, intersections and self-occlusion) and in the 2D image plane (the aforementioned object-background segmentation). Hence when initialising new point tracks, these can be filtered to occupy only the target area. The depth of the 3D features can then be initialised using the intersection between rays from the camera centre and the shape model.

3.4 Final Model Extraction

This section deals with creation of an explicit 3D model, which is the final product of any SfM algorithm. The GP model described above is *implicit* and non-parameteric, *i.e.* while the object's presence/absence can be queried at any point, it has no discrete set of parameters or elements (*e.g.* vertices or edges), and therefore cannot be simply stored for later use, without also storing the entire state of the system. Furthermore, it tends to oversmooth in both interpolated and extrapolated regions. Finally, the canonical GP model cannot be warped according to the mixing coefficients $\boldsymbol{\alpha}$. For these reasons, we produce another model, which is a standard watertight triangular mesh. This model can be provided online, *i.e.* after processing every new frame, however that is usually not required. The triangular model is created using Poisson reconstruction [25,26] on the canonical shape $\mathtt{B}'$. To achieve this, a set of surface normals $\mathcal{N}$ is estimated from the GP model (by sampling points in a very close neighbourhood and fitting a tangent plane), corresponding to every point in $\mathtt{B}'$. The Poisson equation is solved to find the hidden function H, whose gradient approximates these normals

$$\varDelta H = \nabla \mathcal{N}. \tag{9}$$

A collection of smoothed model vertices may then be selected from the H_0 isosurface.

Since the task in NRSfM is reconstruction of *time-varying* shape, the model needs to be non-rigid as well. The transfer of the deformation is achieved as follows. Firstly, a rigid model is created using the canonical shape $\mathtt{B}'$, analogously to the GP model training. Every vertex of the mesh model is assigned a fixed set of features in the cloud, determined as its k nearest neighbours (set to 3 in our experiments). Since for each 3D feature the offset of basis poses (from the canonical shape) is known, the offset of basis poses of each vertex can be computed as a mean of offsets of its k nearest neighbours. Since the topology of the model does not change when performing the warp, its basis shapes differ only by the vertex coordinates: these are computed by applying the offsets to the canonical model.

Finally, the texture of the model is extracted from the image sequence. As we only provide the model at the end, the sequence is processed again in the second pass. The model is warped into the appropriate shape for each frame using previously estimated mixing coefficients, and the texture of visible mesh faces is updated. For every pixel of the texture, we keep a full-covariance normal distribution in the RGB space and the mean is used as the resulting colour. The observer "samples" are weighted according to the observation angle.

4 Experimental Results

4.1 Synthetic Experiments

We perform an initial quantitative evaluation on the synthetic CubicGlobe dataset. This sequence contains a rotating globe which repeatedly warps into cubic shape and then back to sphere. We report performance of the proposed algorithm with a number of quantitative measures, comparing against several state-of-the-art template-free NRSfM techniques which have source code available online. These include BALM [27], using Augmented Lagrange multipliers to solve for the bilinear factorisation problem in the presence of missing data, LIIP [28], using isometric deformation instead of basis shape combination and

SoftInex [12] which employs the material inextensibility prior as a soft constrain in its energy function. These tests measure three important properties of a successful NRSfM technique. Firstly its accuracy of modelling: the fit of the basis shapes to a perfect cube/sphere (the error is expressed relative to the model size, *i.e.* the sphere radius and half of the cube side). The second measured quantity is the accuracy of camera tracking: camera rotation error measured in the angle-axis representation as angular error of both the axis and the rotation angle (since the global coordinate frame is not fixed, the rotation is measured as relative to the first frame). Finally, we measure the depth error of the instantaneous point locations, as Spearman correlation (to overcome the inherent scale ambiguity of the 3D reconstruction) between the measured and ground-truth depth. The sequence will be made available online including all ground truth information, such as shape, trajectory, depth, *etc.*.

It is important to note that all three state-of-the-art comparison methods use the orthographic camera model to simplify computation. This makes it more challenging to evaluate the camera trajectory and depth correlations against the ground truth. To resolve this issue a state of the art Perspective-n-Point algorithm [29] with outlier rejection was used to find the optimal projective camera pose, given the reconstructed 3D point clouds.

Since there are no ground-truth point tracks for this sequence, we provided the state-of-the-art techniques with tracks obtained by our technique. LIIP and SoftInex do not handle occlusions; therefore we run them only on a limited portion of the sequence (the first 50 frames), with only those tracks, which are visible in all 50 frames. Furthermore, the competing approaches do not directly provide meaningful basis shapes. Therefore we use the instantaneous shape from frames 30, 90 and 150 for cube, and 1, 60, 120 and 180 for sphere (where the ground truth shape is pure). The table contains the best possible performance for each of these.

See Table 2 for results. It is clearly visible that BALM failed completely on this sequence, producing large reconstruction and camera rotation errors. Similarly, the depth reported by BALM is not correlated to the GT depth. The results of LIIP are significantly better, with much lower reconstruction errors and rotation error reduced by an order of magnitude, compared to BALM. The average depth correlation is 0.5. SoftInex produces even better 3D reconstructions, with error comparable to the proposed method (although of only one side of the object since it does not handle occlusions). The camera pose is less accurate than that

Table 2. Reconstruction results on the CubicGlobe sequence.

	Cube (%)	Sphere (%)	Axis (°)	Angle (°)	Depth (%)
BALM	51 ± 54	14 ± 12	52.6 ± 28.0	56.9 ± 39.7	5 ± 26
LIIP	29 ± 20	5 ± 5	12.9 ± 20.4	8.1 ± 4.4	50 ± 29
SoftInex	$4 \pm \mathbf{3}$	$\mathbf{3} \pm \mathbf{2}$	22.3 ± 8.5	11.8 ± 8.1	79 ± 13
Proposed	$\mathbf{3} \pm \mathbf{3}$	$\mathbf{3} \pm 3$	$\mathbf{0.4} \pm \mathbf{0.8}$	$\mathbf{3.5} \pm \mathbf{1.4}$	$\mathbf{95} \pm \mathbf{3}$

of LIIP, the reported depth is nevertheless strongly correlated with the ground truth.

The results of SoftInex demonstrate an interesting phenomenon. While the per-frame point depth, returned by the algorithm (and used to infer the non-rigid shape) is realistic, it is "flipped" in the z-direction for some frames (*i.e.* the object side is turned inside out; this is probably due to the lack of temporal smoothness constraint). For a fair comparison, we had to detect and correct this during our experiments. Without this, the results of SoftInex are significantly worse, *e.g.* the depth correlation drops to 16 %. When using the proposed method, the reconstructed models cover the whole object (as visualised in Fig. 1) with very low errors. The camera rotation demonstrates even better performance, with error reduced by an order of magnitude due to its inherent ability to perform tracking and modelling simultaneously. The depth estimated by the proposed method is nearly perfect, reaching 95 % correlation with the observed depth.

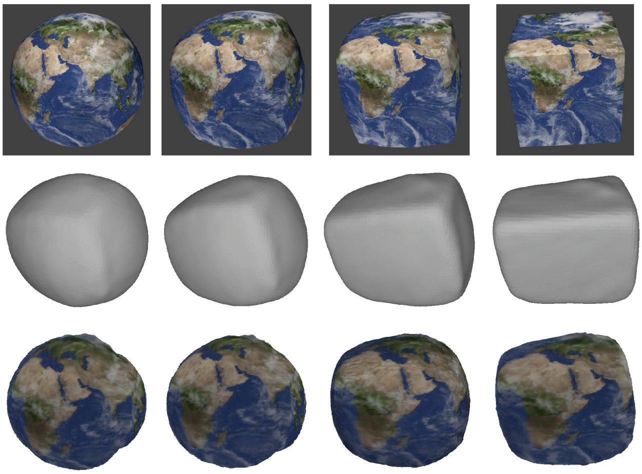

Fig. 1. Example of input sequence and models output by our method: frames #60, 70, 80 and 90 of the sequence CubicGlobe.

See Fig. 3 for visualisation of the obtained mixing coefficients $\boldsymbol{\alpha}^t$ in the first 180 frames of the CubicGlobe sequence. The shape is changing from spherical to cubic linearly, which was closely captured by the coefficient change. Notice the "cropped" peaks, a typical artefact of the proposed method. This is caused by the compactness prior, forcing the basis shapes (spherical and cubical in this case) to lie close to each other and hence being unable to truly capture the very extremes. It, however, does not significantly affect the overall performance, as can be seen in both the qualitative (Fig. 1 and the supplementary material) and quantitative (Table 2) results.

Table 3 shows a breakdown of the execution speed for the different algorithms. It should be reiterated that the competing state-of-the-art techniques use point

request bounding box from user
$\mathcal{S}^1 \leftarrow$ detect initial sparse features
$\mathcal{D}^1 \leftarrow$ initialise dense features
$*\mathcal{L}^1 \leftarrow$ load any supervision features
$\mathtt{B} \leftarrow$ initialise 3D point basis $(\mathcal{S}^1, \mathcal{D}^1, \mathcal{L}^1)$
for $t = 2 \rightarrow T$ **do**
    $\mathcal{S}^t \leftarrow$ track by Lucas-Kanade $(\mathcal{S}^{t-1})$
    $\mathcal{D}^t \leftarrow$ track by dense image registration $(\mathcal{D}^{t-1})$
    $*\mathcal{L}^t \leftarrow$ load any supervision features
    $\mathbf{C}^t \leftarrow$ estimate camera pose $(\mathtt{B}, \mathcal{S}^t, \mathcal{D}^t, \mathcal{L}^t)$
    if $||\mathbf{C}' - \mathbf{C}^t|| > \theta_{\mathbf{C}}$ **then**
        $\mathbf{C}' \leftarrow \mathbf{C}^t$
        Optimise $\mathtt{B}$, $\mathbf{C}^1 ... \mathbf{C}^t$, $\boldsymbol{\alpha}^1 ... \boldsymbol{\alpha}^t$ by BA
        $\mathbf{M} \leftarrow$ retrain shape model $(\mathtt{B}, \boldsymbol{\alpha}^1 ... \boldsymbol{\alpha}^t)$
        $\mathcal{S}^t_{\text{new}} \leftarrow \mathcal{S}^t \cup$ detect sparse features $(\mathbf{M})$
        $\mathcal{D}^t_{\text{new}} \leftarrow \mathcal{D}^t \cup$ detect dense features $(\mathbf{M})$
    end if
    *Create and output explicit mesh model.
end for

Fig. 2. The proposed algorithm overview. Lines marked with * are optional.

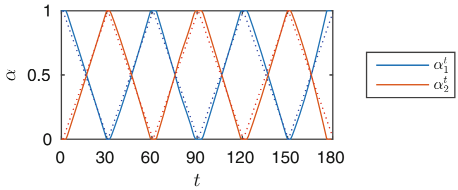

Fig. 3. Mixing coefficients $\boldsymbol{\alpha}^t$ in the CUBICGLOBE sequence (GT shown dotted).

Table 3. Times of processing the first 180 frames of the CUBICGLOBE sequence. The last row does not sum up to 100 % due to various overhead computations, visualisation, I/O wait, *etc.*

	Tracking	**Reconstruction**	**Modelling**
BALM (s)	1 357	248	372
LIIP (s)	1 357	18 768	372
SoftInex (s)	1 357	5 416	372
Proposed (s)	1 357	3 234	372
	22 %	52 %	6 %

tracks provided by the proposed method. Therefore the times for point tracking and model training (necessary for tracking) should be included in their timings for a fair comparison. These are marked in grey. It is also worth noting, that the time for LIIP and SoftInex was consumed in computing reconstruction from only 260 tracks in 50 frames, while the others from nearly 20 000 tracks in 180 frames.

BALM also has scaling issues in terms of memory usage. Operating on the same point tracks used in the proposed approach, BALM consumed more than 200 GB of RAM, two orders of magnitude more than the proposed algorithm.

4.2 Real Data Experiments

To show the performance of the proposed algorithm on real data, we firstly use the recently published 300VW dataset [30–32]. In Fig. 4 we compare the performance of the proposed technique against BALM on the 300VW:002 sequence. When given only the sparse facial landmarks, BALM performs similarly to the proposed technique. However, it has difficulties integrating noisier observations; when BALM is provided with the denser internal trajectories generated by the proposed method, it fails to produce a reasonable reconstruction. In contrast the proposed technique is able to fuse these, to produce a far more detailed reconstruction than from the landmarks alone. Figure 5 shows the resulting

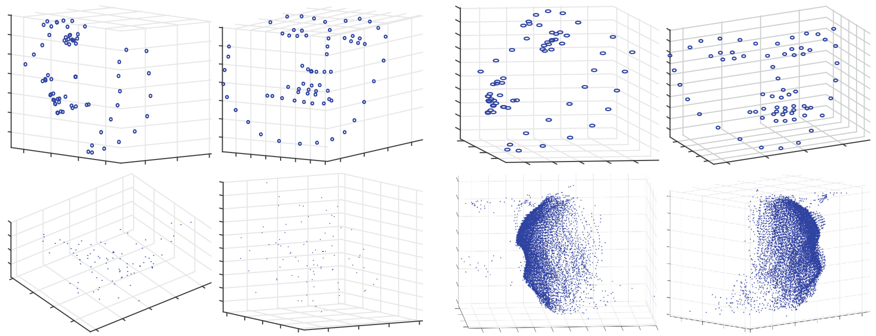

Fig. 4. Comparison of BALM (left two columns) against the proposed technique (right two columns) on the 300VW:002 sequence. Results are shown using only the sparse supervision (top row), and using the sparse supervision with additional densely estimated trajectories (bottom row).

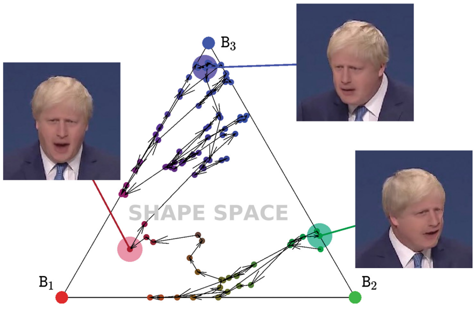

Fig. 5. Reconstructed model overlaid over frames from the 300VW:002 sequence. The shape space visualises the weighted combination of the independent basis shapes. See the supplementary material for an animated version of this figure.

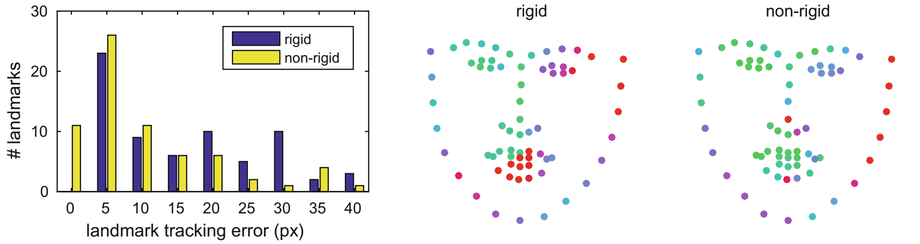

Fig. 6. Landmark tracking error on the 300VW:002 sequence, when using rigid and non-rigid tracking and reconstruction. Left: error histogram, right: landmark error from low (green) to high (red). (Color figure online)

reconstruction of our technique and the trajectory of the model in the shape space defined by B.

It is not only the reconstruction which benefits from the proposed joint approach. Using a non-rigid model can significantly improve tracking results as well. This is demonstrated in Fig. 6, where results are compared between rigid and non-rigid 3D tracking. For non-rigid objects, a "centre" is ill defined. Therefore, a face-tracking scenario is used and the accuracy of landmark tracking is measured. The error is defined as the distance between the GT and the landmarks tracked using the non-rigid 3D model. For each landmark, the error is averaged over all frames. It can be seen that the proposed method has a fraction of landmarks tracked with near-zero error, while the rigid case has no "perfectly tracked" landmarks. Additionally, the rigid variant has a significant portion of landmarks tracked with errors around 20–30 px (mostly near the mouth where the non-rigid deformation is the most pronounced). On average, the tracking error is reduced from 17.4±14.1 to 10.8±10.5 px by using a non-rigid model.

In Fig. 7 we explore the performance of the proposed technique in the fully unsupervised scenario, on the FACE [17] and T-SHIRT [33] sequences. See the

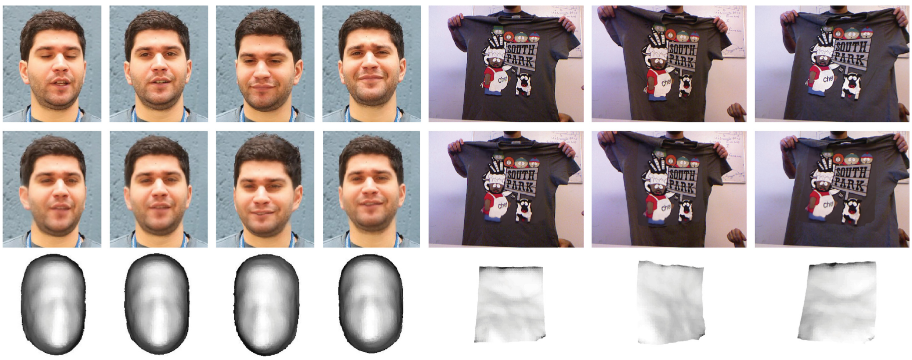

Fig. 7. Example of modelling results on FACE [17] (left) and T-SHIRT [33] (right). From top to bottom: original video frames; video frames overlaid with the instantaneous models; the instantaneous model on its own (untextured).

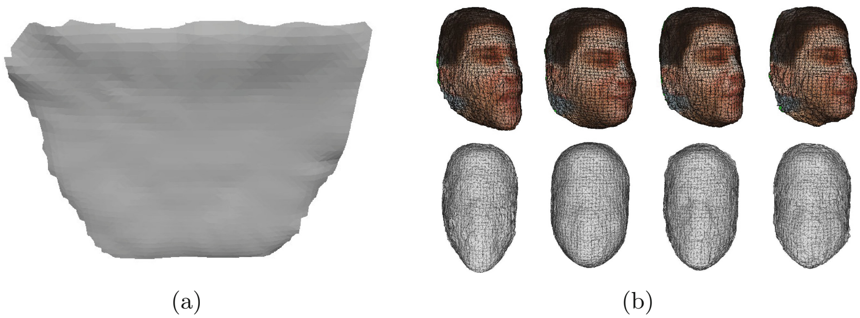

Fig. 8. (a) Details of the model obtained (directly) from the T-SHIRT sequence [33]. (b) Basis shapes obtained from the FACE sequence [17].

Table 4. Quantitative results on the T-SHIRT sequence.

	PCA [33]	Uncon. LVM [33]	CLVM [33]	DDD [18]	Proposed
Error (mm)	18.44	15.50±1.78	14.79±0.90	7.05	17.82±4.72

supplementary material for resulting videos. The models generated by our approach are similar to the results generated by state-of-the-art NRSfM techniques. However, it should be re-emphasised that we solve a much more challenging problem: the fully unsupervised online scenario. As can be seen, all estimated target poses are feasible and the estimated shapes model the deformations well despite the lack of supervision. It is also obvious from the second row that estimates of the rigid motion (*i.e.* the camera pose) are extremely accurate. In Fig. 8a, the canonical T-SHIRT model (before cropping to contain only the region of interest) is shown in detail. Notice the creases near the top of the model, caused by the way the t-shirt is held. Figure 8b shows the basis shapes automatically identified by our method and used in the reconstruction shown in Fig. 7 (with a wireframe mesh overlaid to help visualise the 3D shape). Finally, Table 4 brings quantitative comparison on the T-SHIRT sequence. The results indicate the proposed approach is competitive with state of the art, even though it does not use a template or another kind of prior knowledge and operates directly on the raw RGB images.

5 Summary

In this paper, a novel NRSfM algorithm was introduced. Its main advantage over conventional NRSfM approaches is that it requires no external supervision (e.g. pre-computed clean point tracks): everything required is computed directly from the input video and the only external input is the target selection by a bounding box in the first frame. It is able to autonomously create 3D models from unseen video-sequences. The proposed algorithm is more generic than state-of-the-art methods, with trivial extension to different camera models and additional priors, constraints and regularisations. In addition, it removes several important limitations of conventional methods, most importantly it provides robustness against strong target rotation and self-occlusion.

One of the limitations of the approach is the assumption that the object is roughly compact. Therefore the model is unable to capture more complicated shapes such as walking humans. This is however a limitation of virtually all current model-free approaches.

Acknowledgements. This work was supported by the EPSRC project EP/I011811/1: "Learning to Recognise Dynamic Visual Content from Broadcast Footage" and the SNSF Sinergia project "Scalable Multimodal Sign Language Technology for Sign Language Learning and Assessment" (SMILE) grant agreement number CRSII2 160811.

References

1. Agudo, A., Agapito, L., Calvo, B., Montiel, J.: Good vibrations: a modal analysis approach for sequential non-rigid structure from motion. In: CVPR (2014)
2. Agudo, A., Montiel, J., Agapito, L., Calvo, B.: Online dense non-rigid 3D shape and camera motion recovery. In: BMVC (2014)
3. Paladini, M., Bartoli, A., Agapito, L.: Sequential non-rigid structure-from-motion with the 3d-implicit low-rank shape model. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6312, pp. 15–28. Springer, Heidelberg (2010). doi:10.1007/978-3-642-15552-9_2
4. Tao, L., Matuszewski, B.J.: Non-rigid structure from motion with diffusion maps prior. In: CVPR (2013)
5. Lebeda, K., Hadfield, S., Bowden, R.: 2D or Not 2D: bridging the gap between tracking and structure from motion. In: Cremers, D., Reid, I., Saito, H., Yang, M.-H. (eds.) ACCV 2014. LNCS, vol. 9006, pp. 642–658. Springer, Heidelberg (2015). doi:10.1007/978-3-319-16817-3_42
6. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: a factorization method. Int. J. Comput. Vis. **9**, 137–154 (1992)
7. Bregler, C., Hertzmann, A., Biermann, H.: Recovering non-rigid 3D shape from image streams. In: CVPR (2000)
8. Dai, Y., Li, H., He, M.: A simple prior-free method for non-rigid structure-from-motion factorization. In: CVPR (2012)
9. Rabaud, V., Belongie, S.: Linear embeddings in non-rigid structure from motion. In: CVPR (2009)
10. Bartoli, A., Gay-Bellile, V., Castellani, U., Peyras, J., Olsen, S., Sayd, P.: Coarse-to-fine low-rank structure-from-motion. In: CVPR (2008)
11. Perriollat, M., Hartley, R., Bartoli, A.: Monocular template-based reconstruction of inextensible surfaces. Int. J. Comput. Vis. **95**, 124–137 (2011)
12. Vicente, S., Agapito, L.: Soft inextensibility constraints for template-free non-rigid reconstruction. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7574, pp. 426–440. Springer, Heidelberg (2012). doi:10.1007/978-3-642-33712-3_31
13. Agudo, A., Calvo, B., Montiel, J.: Finite element based sequential bayesian non-rigid structure from motion. In: CVPR (2012)

14. Eriksson, A., van den Hengel, A.: Efficient computation of robust low-rank matrix approximations in the presence of missing data using the L1 norm. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2010)
15. Zollhofer, M., Niessner, M., Izadi, S., Rehmann, C., Zach, C., Fisher, M., Wu, C., Fitzgibbon, A., Loop, C., Theobalt, C., Stamminger, M.: Real-time non-rigid reconstruction using an RGB-D camera. ACM Trans. Graph. (TOG) **33**, 156 (2014)
16. Newcombe, R., Fox, D., Seitz, S.: DynamicFusion: reconstruction and tracking of non-rigid scenes in real-time. In: CVPR (2015)
17. Garg, R., Roussos, A., Agapito, L.: A variational approach to video registration with subspace constraints. Int. J. Comput. Vis. **104**, 286–314 (2013)
18. Yu, R., Russell, C., Campbell, N.D.F., Agapito, L.: Direct, dense, and deformable: template-based non-rigid 3D reconstruction from RGB video. In: ICCV (2015)
19. Strasdat, H., Montiel, J., Davison, A.: Real-time monocular SLAM: why filter? In: ICRA (2010)
20. Weinzaepfel, P., Revaud, J., Harchaoui, Z., Schmid, C.: DeepFlow: large displacement optical flow with deep matching. In: ICCV (2013)
21. Torr, P., Zisserman, A.: MLESAC: a new robust estimator with application to estimating image geometry. Comput. Vis. Image Underst. **78**, 138–156 (2000)
22. Agarwal, S., Mierle, K., et al.: Ceres solver. http://ceres-solver.org
23. Paladini, M., Del Bue, A., Stosic, M., Dodig, M., Xavier, J., Agapito, L.: Factorization for non-rigid and articulated structure using metric projections. In: CVPR (2009)
24. Lebeda, K., Hadfield, S., Bowden, R.: Dense rigid reconstruction from unstructured discontinuous video. In: ICCV 3DRR (2015)
25. Kazhdan, M., Bolitho, M., Hoppe, H.: Poisson surface reconstruction. In: SGP (2006)
26. Kazhdan, M., Hoppe, H.: Screened poisson surface reconstruction. ACM Trans. Graph. (TOG) **32**, 29 (2013)
27. Del Bue, A., Xavier, J., Agapito, L., Paladini, M.: Bilinear modeling via augmented lagrange multipliers (BALM). TPAMI (2012)
28. Chhatkuli, A., Pizarro, D., Bartoli, A.: Non-rigid shape-from-motion for isometric surfaces using infinitesimal planarity. In: BMVC (2014)
29. Ferraz, L., Binefa, X., Moreno-Noguer, F.: Very fast solution to the PnP problem with algebraic outlier rejection. In: CVPR (2014)
30. Chrysos, G., Antonakos, E., Zafeiriou, S., Snape, P.: Offline deformable face tracking in arbitrary videos. In: ICCVW (2015)
31. Shen, J., Zafeiriou, S., Chrysos, G., Kossaifi, J., Tzimiropoulos, G., Pantic, M.: The first facial landmark tracking in-the-wild challenge: Benchmark and results. In: ICCVW (2015)
32. Tzimiropoulos, G.: Project-out cascaded regression with an application to face alignment. In: CVPR (2015)
33. Varol, A., Salzmann, P., Urtasun, R.: A constrained latent variable model. In: CVPR (2012)