Intrinsic Characterization of Dynamic Surfaces

Tony Tung Takashi Matsuyama
Graduate School of Informatics, Kyoto University, Japan
{tung,tm}@vision.kuee.kyoto-u.ac.jp

Abstract

This paper presents a novel approach to characterize deformable surface using intrinsic property dynamics. 3D dynamic surfaces representing humans in motion can be obtained using multiple view stereo reconstruction methods or depth cameras. Nowadays these technologies have become capable to capture surface variations in real-time, and give details such as clothing wrinkles and deformations. Assuming repetitive patterns in the deformations, we propose to model complex surface variations using sets of linear dynamical systems (LDS) where observations across time are given by surface intrinsic properties such as local curvatures. We introduce an approach based on bags of dynamical systems, where each surface feature to be represented in the codebook is modeled by a set of LDS equipped with timing structure. Experiments are performed on datasets of real-world dynamical surfaces and show compelling results for description, classification and segmentation.

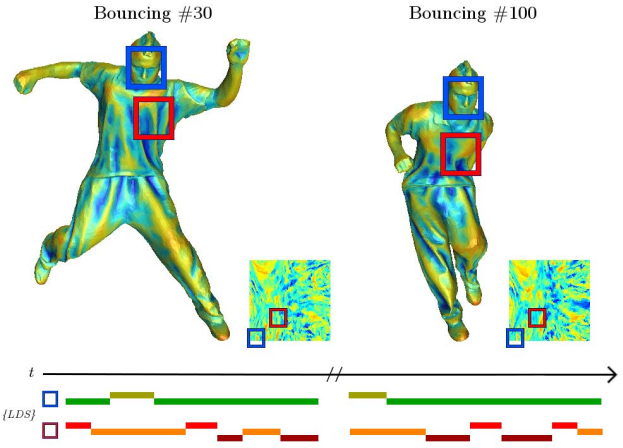

Figure 1. Dynamic surface characterization from intrinsic property extraction, tracking, and dynamics modeling across time. Here, we show local curvatures estimated at each surface point, and curvature maps obtained after mapping on square domain. Surface regions (e.g., colored squares) can be classified using curvature dynamics modeled by sets of dynamical systems {LDS}.

1. Introduction

Since several decades computer vision technologies have provided various solutions for scene understanding using global shape and appearance of objects (e.g., face detection, pose estimation, action recognition, etc.). Nowadays, advances in visual sensing systems (for color and depth) allow us to capture smaller variations and details on object surfaces in real-time (i.e., high resolution at high frame rate). For example, techniques such as performance capture or 3D video [27, 17, 34, 11, 20] can return complete and accurate 3D dynamic surface models, reconstructed by multiview stereo (MVS) methods or fusion of depth maps. Hence, it is now possible to increase the understanding level by exploiting local geometry information. Tackling this problem can potentially help to overcome many issues caused by appearance inconsistency that affect general computer vision and pattern recognition algorithms. Here, we propose to characterize, classify and segment dynamic deformable surfaces using surface intrinsic property dynamics (see Fig. 1).

Dynamic surfaces representing real-world objects (e.g., humans, soft tissue organs, fluids, etc.) can be assumed as a stream of temporally continuous and indefinitely varying 3D geometrical data that possess certain temporal statistics. For example, clothing made of soft fabrics worn by a human in motion usually exhibit more surface variations that bare skin. Unfortunately, to capture those complex variations one cannot (only) rely on visual appearance-based methods [13, 7, 30], as surface texture of complete 3D surface models can be poor (e.g., skin, solid color clothing, etc.) and is usually subject to color inconsistencies due to the different lighting conditions from multiple capture viewpoints. On the other hand, geometry is subject to reconstruction artifacts (caused by occlusion, sensor noise, resolution, etc.) and has therefore limited accuracy. However, actual sensing devices and capture systems can already provide data which are good enough for research and many applications, and it is reasonable to assume that reconstruction accuracy and robustness will continue to improve very quickly. Hence, we propose to characterize dynamic surfaces using a

geometry-dynamics-based approach that relies on intrinsic surface properties as follows: (1) surfaces are first aligned in order to locate and track surface feature (e.g., local curvature) variations over time, (2) temporal variations are then modeled using several linear dynamical systems (LDS) per feature to capture both spatiotemporal variations and state changes, and (3) timing structure of LDS are introduced into bags of dynamical systems (BoS) that are used for description and classification of surface regions (see Fig. 1).

The rest of the paper is organized as follows. The next section discusses work related to the techniques presented in this paper. Section 3 presents the extraction dynamic surface intrinsic feature. Section 4 introduces the LDS and timing structure models. Section 5 describes surface dynamics modeling using bags of dynamical systems. Section 6 shows experimental results. Section 7 concludes with a discussion on our contributions.

2. Related work

Complete reconstruction of dynamic surfaces is an active research area due to the numerous potential applications: medicine, sports, entertainment, digital archiving, etc. During the last decade, several multiview video systems and applications have been developed [22, 27, 17, 10, 2, 34, 11, 18, 19, 20, 37]. They are able to capture real-world human or animal performances, and produce free-viewpoint video of the subjects in motion in a virtual world (see Fig. 2). Usually, several sensing devices are spaced around a scene (e.g., in a studio) and synchronously perform the capture . The devices can either be a set of calibrated video cameras, or even handheld depth cameras. Additionally, 3D laser scanner can be used to leverage the reconstruction accuracy. With these techniques, subjects are captured without wearing any special markers, as opposed to motion capture methods (mocap). The resulting performance capture or 3D video consists of a stream of textured surface mesh models undergoing free-form deformation.

Traditionally every frame is reconstructed independently, and consecutive meshes have inconsistent connectivity (and topology). However, recent efforts have been done to produce consistent sequences by 3D scene flow estimation, surface matching or tracking [39, 33, 12, 38, 41, 4, 28, 35, 16, 3]. Nevertheless, photometric feature matching approaches require surface models with good texture and color consistency between the multiple capture viewpoints, and across time. Hence, most appearance-based methods are not able to accurately track true deformations of low-frequency surface details (e.g., wrinkles on solid color clothing, etc.).

We propose to model complex surface variations using linear dynamical systems (LDS). LDS are a generalization of Hidden Markov Models (HMM) [29] where the underlying state-space is continuous instead of discrete. In particular, dynamical models have been applied in computer vision for dynamic texture modeling [32, 13], recognition[32, 30], and segmentation[14, 7, 40]. And as well for facial movement synchronization [24], human action recognition [9], etc. In [30], the authors propose to tackle challenging scenarios and model dynamic textures with a collection of LDS, by following the bag of features (BoF) approach, where a LDS is associated to a spatiotemporal volume obtained by tracking a feature point. However, in the context of dynamic surfaces from human performance, the nature of deformations can be heterogeneous in time, and therefore requires several LDS for modeling. It is then necessary to take into account the timing structure of LDS. Thus, dynamic surfaces can be segmented into patches that are classified into regions corresponding to each body parts (e.g., head, upper-body, arms, etc.). To the best of our knowledge, no prior work has attempted to tackle this problem.

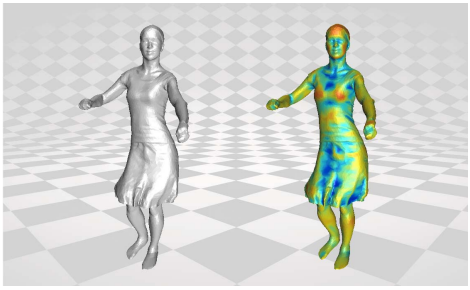

Figure 2. Dynamic surfaces reconstructed from multiview stereo methods (MVS) [2]: surface and processed surface (curvatures).

3. Dynamic surface feature extraction

This section presents surface intrinsic feature extraction from surface points which are tracked across time. In particular, we estimate local curvatures as features using a continuous surface shape index.

3.1. Surface intrinsic characterization

To perform surface intrinsic characterization, we propose to represent local curvatures by computing the Koenderink shape index for each surface point, as it is known to be more stable for natural scenes than a classification by Gaussian and mean curvatures [25]. The shape index describes the local type of a shape as a continuous parameter. The differential structure of a surface can be captured by the local Hessian matrix $\mathcal{H}$, which is computed using surface normals:

$$\mathcal{H} = \begin{pmatrix} -\left(\frac{\partial \mathbf{n}}{\partial x}\right)_x & -\left(\frac{\partial \mathbf{n}}{\partial x}\right)_y \\ -\left(\frac{\partial \mathbf{n}}{\partial y}\right)_x & -\left(\frac{\partial \mathbf{n}}{\partial y}\right)_y \end{pmatrix}, \qquad (1)$$

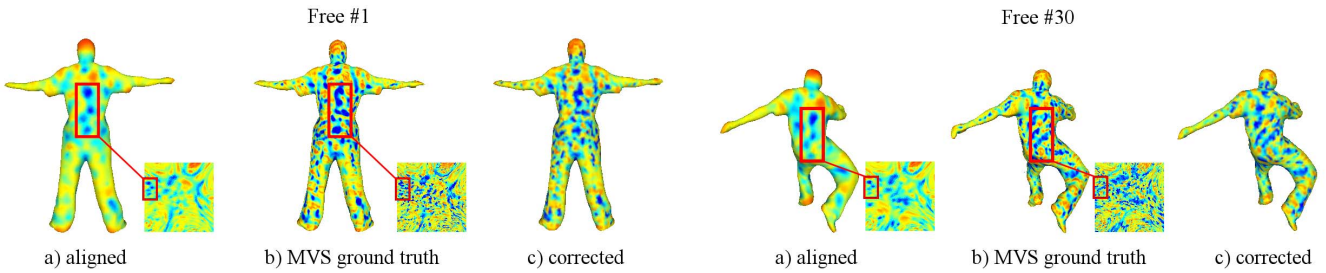

Figure 3. Local curvature variation across time: a) from surface alignment [4], b) from MVS ground truth [34] and c) from alignment and correction. Although global geometry is well preserved by [4], local geometry information is lost across time (see boxes).

where $\mathbf{n}$ is a surface normal, and which eigenvalues are the principal curvatures κ_1 and κ_2 ($\kappa_1 \geq \kappa_2$). For all surface point, the shape index σ describes local surface topology in terms of the principal curvatures:

$$\sigma = \frac{2}{\pi} \arctan \frac{\kappa_2 + \kappa_1}{\kappa_2 - \kappa_1}. \quad (2)$$

The values of $\sigma \in [-1, 1]$ encode the type of curvature such as: cup, rut, saddle rut, saddle ridge, ridge, dome, cap, etc. Local curvatures computed on different surface meshes are shown in Fig. 1, 2 and 3. Cups are in blue and caps in red.

3.2. Intrinsic feature tracking

One challenge to overcome when characterizing surface dynamics is surface alignment for surface point tracking. As discussed in Sect. 2, methods involving color information cannot be used for that purpose as surfaces from performance capture (or 3D video data) usually suffer from color inconsistency or poorly textured regions. As well, methods which are too sensitive to surface deformation or topology change can produce inaccurate results. Here, we propose to use [4] to perform surface alignments independently from color information and topology change. Nevertheless, while the global surface geometry is correctly deformed and aligned across time, the patch-based approach does not preserve intrinsic information such as local curvatures. Hence, we propose to register original surface meshes (with computed local curvature information at full resolution) to sequences aligned as in [4], and correct local curvature with exact values for each mesh vertex on the latter ones by assigning the nearest neighbor values. Actually, 3D video sequences obtained from MVS usually contain surface noise. However, as the reconstruction is performed frame-by-frame they can still be a good approximation of ground truth surface as no noise is propagated through the sequence, as opposed to spatiotemporal reconstruction. Figure 3 shows local curvatures computed on surface mesh models across time. Curvature maps obtained after surface alignment and mapping on square parametrization domain [31] are given for visualization purpose. Note that recently in [36], the authors have proposed an invariant surface descriptor that could potentially be used for surface alignment and surface point tracking.

4. Dynamic surface modeling using LDS

When representing dynamic surfaces as curvature maps (see Fig. 1 and 3), analogy can be made with dynamic textures [32]. However, surfaces from performance capture can exhibit heterogeneous deformations in time (see Sect. 2). Hence we model surface dynamics using hybrid linear dynamical systems (hybrid LDS) that can describe both continuous and discrete events. The model consists of a two-layer architecture: (1) a set of N LDS $\mathcal{D} = (D_1 \dots D_N)$ to model complex continuously changing events, and (2) a finite state machine (FSM) that represents states and state transitions (i.e., duration and temporal relationship).

4.1. Hybrid linear dynamical system

Assuming a temporal sequence of an observed signal $Y = \{y(t)\}_{t\geq 0}$, $y(t) \in \mathbf{R}^m$, and its hidden states $X = \{x(t)\}_{t\geq 0}$, $x(t) \in \mathbf{R}^n$ belonging to a continuous state space, a linear dynamical system D_i can be defined as:

$$\begin{cases} x(t+1) &= A_i x(t) + g_i + v_i(t) \\ y(t) &= Cx(t) + w(t), \end{cases} \quad (3)$$

where $A_i \in \mathbf{R}^{n\times n}$ is the state transition matrix which models the dynamics of D_i, g_i is a bias vector and $C \in \mathbf{R}^{m\times n}$ is the observation matrix which maps the hidden states to the output of the system by linear projection. $v_i(t) \sim \mathcal{N}(0, Q_i)$ and $w(t) \sim \mathcal{N}(0, R)$ are process and measurement noises modeled as Gaussian distributions with null averages and Q_i and R covariances respectively. Particularly $(A_i, C) \in \mathbb{GL}(n) \times \mathbb{ST}(m, n)$, where $\mathbb{GL}(n)$ is the group of invertible matrices of size n, and $\mathbb{ST}(m, n)$ is the Stiefel manifold. Eq. 3 is known for its ability to model complex spatiotemporal variations (e.g., for dynamic textures [13, 30], human actions [9]). For heterogeneous scenes or patterns, mixture

of LDS are used, and the number N of LDS and all the LDS parameters can be estimated using training datasets and optimized by Expectation-Maximization (e.g., for dynamic texture segmentation [7], and facial movement recognition [24]).

In order to model the system state changes, the set of LDS $\mathcal{D}$ is represented by a finite state machine (FSM). The FSM consists of a discrete set of states $\mathcal{Q} = \{q_i\}_{i=1}^{N}$, where each q_i corresponds to a LDS D_i. Hence, as $\{q_i\}$ (and therefore D_i) is activated, a sequence of continuous states $\{x(t)\}$ is generated and mapped to the output observation space as $\{y(t)\}$. $\{y(t)\}$ can then be entirely modeled by a set of N LDS, and the state timing structures given by the FSM layer.

4.2. Patch-based spatiotemporal description

As surfaces are represented by (polygonal) meshes, numerical approximations have to be handled. Moreover, drift effects inherent to [4] should to be compensated (especially for long sequences). Hence, we propose to consider one spatiotemporal descriptor per patch, where a patch consists of a set of mesh vertices. In our implementation, patches are as in [4] (e.g., the sequence *Bouncing* [11, 4] contains 450 patches having 4 to 13 vertices each), and we model each patch using N LDS.

Figure 4 shows signals $\{y(t)\}$ in the observation space, representation of hybrid LDS model with $N = 4$ LDS using intervals, and generated signals from the model. Here, each interval is described by a state q_i of the FSM (with a unique color) and a duration $\tau_j \geq 0$.

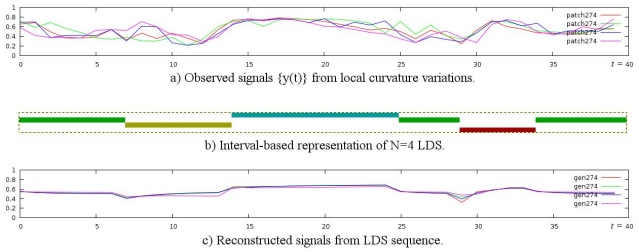

Figure 4. Hybrid LDS modeling. a) Observed signals from surface patch #274 in Bouncing sequence [4] (torso region). b) The intervals represent the timing structure (i.e., state transitions and durations) given by hybrid LDS modeling. (Each LDS is represented by a unique color.) c) Reconstructed signals from LDS sequence.

5. Dynamic surface characterization

Bag-of-features (BoF) have been successfully applied to various visual classification tasks thanks to their ability to capture invariance aspects of local features [26, 21, 30]. In [30], the authors introduce the bags of dynamical systems (BoS) for dynamic texture recognition and outperforms [32]. Here, we propose to apply the BoS framework to characterize dynamic surfaces. Moreover, each surface patch is modeled by a set of N LDS (as opposed to only one per video feature in prior work). As well, we introduce timing structure information given by the hybrid LDS model in the codebook formation of BoS.

5.1. Codebook generation

As in the BoF framework, the codebook is generated by clustering extracted features from a training dataset to obtain the codewords that form a dictionary. Here, our features are sets of LDS parameters (extracted from surface patches) belonging to a non-Euclidean space: $(A_i, C) \in \mathbb{GL}(n) \times \mathbb{ST}(m, n)$ (see Sect. 4). Hence, clustering algorithms used in the Euclidean space (such as k-means) cannot be applied directly, as discussed in prior work [7, 5, 6, 1].

Distance between LDS. Several methods were proposed in order to compare LDS, based on Kullback-Leibler divergence or Martin distance (see [32, 13, 30, 24]). The Martin distance between two LDS parameters $M_1 = (A_1, C_1)$ and $M_2 = (A_2, C_2)$ is based on the subspace angles between the two systems M_1 and M_2 belong to. The subspace angles $\{\theta_i\}_{i=1}^{n}$are defined as the principal angles between *observability subspaces* [32], and can be obtained by solving the following Lyapunov equation for $\mathcal{P}$:
$\mathcal{A}^\top \mathcal{P} \mathcal{A} = -\mathcal{C}^\top \mathcal{C}$, where

$$\mathcal{P} = \begin{pmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{pmatrix} \in \mathbb{R}^{2n \times 2n}, \mathcal{A} = \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix} \in \mathbb{R}^{2n \times 2n},$$

$$\mathcal{C} = \begin{pmatrix} C_1 & C_2 \end{pmatrix} \in \mathbb{R}^{m \times 2n}, \tag{4}$$

and $\cos^2 \theta_i = i$-th eigenvalue of $(P_{11}^{-1} P_{12} P_{22}^{-1} P_{21})$. The Martin distance d_M between M_1 and M_2 is then given by

$$d_M(M_1, M_2) = -\ln \prod_{i=1}^{n} cos^2 \theta_i. \tag{5}$$

Clustering of LDS. As we are using the BoS framework, we employ an approximate averaging method as in [30]. Let us denote $D \in \mathcal{R}^{T \times T}$ the pairwise dissimilarity matrix between all features obtained using the Martin distance d_M, where $T = N \times \#\{\text{features}\}$. In [30], the authors propose to embed all features in a lower dimension space where k-means clustering can be applied using Euclidean distance (Multi Dimensional Scaling is applied using D).

In our framework, as each surface region is represented by a limited number of LDS features $\{(A_i, C)\}$, T remains relatively small. Hence, to improve the robustness to noise and outliers of the clustering (compared to k-means), we propose to use the k-medoids algorithm [23] in the LDS parameter space, where minimizations are computed from sums of pairwise dissimilarities using the Martin distance

(i.e., using D). By definition, a medoid is the feature of a cluster, whose average dissimilarity to all the features in the cluster is minimal. It is a most centrally located point in the cluster. The clustering of T features $\{x_j\}_{j=1}^{T}$ into a set of K clusters $\mathcal{S} = \{S_k\}_{k=1}^{K}$ can be achieved by finding the set of K medoids $\{m_1, ..., m_K\}$ using the following steps:

1. Initialization (i=0): random selection of K medoids $\{m_k^{i=0}\}_{k=1}^{K}$ among the T features $\{x_j\}_{j=1}^{T}$.

2. Associate each x_j to the nearest cluster $S_k^{(i)}$: $S_k^{(i)} = \{x_j : d_M(x_j, m_k^{(i)}) \leq d_M(x_j, m_{k'}^{(i)}), \forall k' = 1...K\}$,

3. Select the new set of medoids that minimize the distances between features in each cluster:

$$\xi(x_j, S_k^{(i)}) = \sum_{x_k \in S_k^{(i)} \setminus \{x_j\}} d_M(x_j, x_k), \quad (6)$$

$$m_k^{(i+1)} = \arg\min_{x_j \in S_k^{(i)}} \left(\xi(x_j, S_k^{(i)}) \right), \quad (7)$$

   where $\xi(x_j, S_k^{(i)})$ represents the *cost* of assigning x_j as a medoid of $S_k^{(i)}$.

4. Repeat 2 - 3 until convergence (i.e., $m_k^{(i+1)} = m_k^{(i)}$).

To overcome the dependence to initialization, the clustering is run several times and the configuration that returns the minimal total cost over all clusters is selected:

$$\{F_1, ..., F_K\} = \arg\min_{\{\{m_1,...,m_K\}\}} \left(\sum_{k=1}^{K} \xi(m_k, S_k) \right), \quad (8)$$

where $\{F_1, ..., F_K\}$ represents the set of K codewords that forms the vocabulary of the codebook. This strategy, instead of selecting the most frequent clusters, returns more homogeneous clusters regarding size and spatial arrangement [23].

5.2. Soft-weighting with term frequency

Let us consider all the features $\{x_j\}$ extracted from a dynamic surface (i.e., all the sets of LDS from each patch). In the BoF framework, each feature contributes to a set of weights $\{w_1, ..., w_K\}$ associated to codewords $\{F_1, ..., F_K\}$ that represents the object (e.g., for classification). We propose to use *soft-weighting* as it is less sensitive to noise [21, 30], and we introduce timing structure information given by the hybrid LDS modeling into the weighting scheme using the *term frequency* ρ_j:

$$w_k = \sum_{i=1}^{N_0} \sum_{j=1}^{M_i} \left[(\alpha + \beta * \rho_j) \frac{1}{2^{i-1}} \text{sim}(x_j, F_k) \right], \quad (9)$$

where $N_0 = 4$ is the number of top-nearest codewords to be considered for each x_j, M_i represents the number of features whose i-th nearest neighbor is F_k, and

$$\text{sim} = 1 - \frac{d_M}{\max(D)} \quad (10)$$

is a similarity measure between LDS, where $\max(D)$ is the biggest element of D (see definition of D above). Finally, the contribution of feature x_j is weighted as well by

$$\rho_j = \frac{N_j}{N_{tot}}, \quad (11)$$

where N_j is the total duration of the state (i.e., sum of interval lengths) represented by x_j with respect to the total duration N_{tot} of a data sample, and $\alpha < 1$ and $\beta < 1$ are weighting factors (e.g., $\alpha = 0.7$ and $\beta = 0.3$). If $\beta = 0$, then we lose the timing structure characterizing the duration of each state of the LDS in the model.

5.3. Classification

To compare and classify the codewords, we use Support Vector Machines (SVM) with Radial Basis Function (RBF) kernel

$$K(x, y) = \exp^{-\gamma d(x,y)}, \quad (12)$$

where γ is a free parameter that can be learnt by cross-validation, and $d(W_1, W_2)$ is a distance on the histogram space, such as the χ^2:

$$d_{\chi^2}(x, y) = \sum_i \frac{(x_i - y_i)^2}{x_i + y_i}. \quad (13)$$

As well, we obtained good performance using distance RBF kernels that are Laplacian and sub-linear:

$$d(x_i, y_i) = \sum_i |x_i - y_i|^b, \quad \text{with} \quad b < 2, \quad (14)$$

which are popular in image retrieval and satisfy the Mercer's condition [15]. The SVM show more stability than $k-$Nearest Neighbor ($k-$NN) in our experiments.

6. Experimental results

Discussion on real-world datasets. To evaluate the proposed model, we use publicly available datasets of 3D video sequences reconstructed from real human performances from the University of Surrey [34], INRIA Grenoble [2], and MIT CSAIL [11]. (Sequences from [37] were not used in the current evaluation as the reconstruction accuracy was not sufficient.) These sequences represent subjects wearing loose clothing (e.g., T-shirt) while turning, dancing, and/or jumping. Every mesh has reasonable resolution and quality which allow us to capture local surface variations across

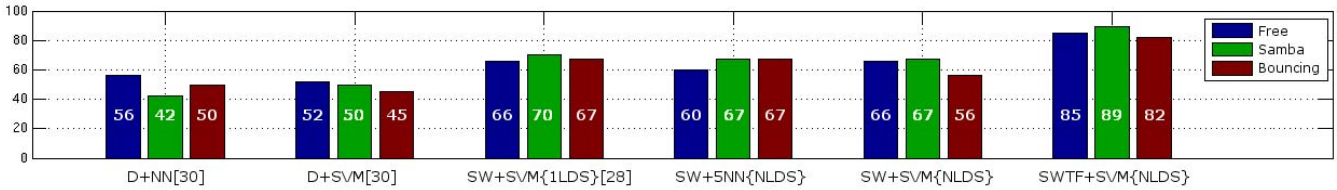

Figure 5. Classification results for the different sequences using different approaches. Our method using BoS with SW and timing structure and SVM returns the best performance compare to the state-of-the-art techniques used for dynamic texture recognition.

time. However, using intrinsic characterization, we could observe that reconstructed surfaces from [34] (such as Free) are still very noisy at small scale due to drawbacks from MVS, despite being visually very compelling. On the other hand, surfaces from [11] and [2] (such as Samba and Bouncing respectively) contain drawbacks for spatiotemporal reconstruction, and therefore reconstructed surfaces of clothing might eventually be more rigid and less prone to wrinkle. However all of these dynamic surfaces could exhibit temporal statistics in different regions or body parts. In what follows, we particularly focus on rigid and non-rigid regions as for the time being, it is still difficult to characterize different surface materials based only on surface dynamics. For example, in the Samba dataset, the subject wears a dress that moves during the dance. However, the top of the dress is tight and its surface does not exhibit that much variations. Conversely, in the Bouncing sequence, the subject wears large T-shirt and pants that exhibit lots of variations during the jumps. However, it is challenging to distinguish the T-shirt from the pants using surface dynamics as variations are similar (and may have been kept rigid and smooth by the reconstruction process).

Baseline for comparison. As discussed in Sect. 4, dynamic surfaces can be treated as dynamic textures although surface variations from performance capture can be unpredictable or heterogeneous in time. As no prior work on surface dynamics characterization as been proposed in the computer vision literature, we propose to use state-of-the-art approaches related to dynamic texture recognition as baseline for comparison. In [8, 32], the authors use a single LDS to model a video sequence (of dynamic textures), the Martin distance is used to calculate distances between LDS, and NN and SVM are used for classification. In [30], the authors use BoS with one LDS per video feature, soft-weighting (SW) and SVM for classification. We abbreviate these approaches D+NN, D+SVM and SW+SVM{1LDS} respectively. As well, our dynamic surface models are abbreviated SW+5NN{NLDS}, SW+SVM{NLDS}, and SWTF+SVM{NLDS} for classification using BoS with SW and $k-$NN classifier ($k = 5$), using BoS with SW and SVM, and using BoS with SW and timing structure and SVM, respectively.

Dynamic surface classification. As we deal with dynamic surfaces representing continuous human performances, we expect to characterize repetitive patterns that can be found in surface deformations. Challenges come from surface noise and irregular (repetition of) patterns, as subjects repeat or perform various actions in a same sequence. We could observe that even *rigid* surface regions such as bare skin or faces exhibit some variations.

First, we propose to classify dynamic surface patches from different sequences into *rigid* and *non-rigid* classes. Patches are extracted from aligned surface sequences [4]. Furthermore, to obtain ground truth classification, we manually labeled each patch and assigned them to a surface region (i.e., body part) that belongs to either the *rigid* class or the *non-rigid* class. The labeling process is by far the most time consuming (and tedious) step when preparing data for learning. For example, Free has 514 patches divided into 20 subregions (such as right forearm, left forearm, head, front torso, back torso, etc.), Samba has 361 patches divided into 9 subregions, and Bouncing has 450 patches divided into 9 subregions.

During our experiments, best results were obtained with $N = 4$ LDS in the hybrid LDS modeling with LDS order $n = 6$ and $K = 8$ codewords. Sequence Free contains 499 frames, and sequences Samba and Bouncing contain both 174 frames. For the Free sequence, frames #1:99 are used for training, while frames #100:199, #200:299, #300:399 and #400:499 are used for testing. For Samba and Bouncing sequences, frames #1:59 are used for training, while frames #60:99, #100:139 and #140:174 are used for testing. In figure 5, we present classification results for the different sequences using the different methods described above. In general, the classification tasks are more difficult for the approaches D+NN, D+SVM that use a single LDS [8, 32]. Our method using BoS with SW and timing structure and SVM returns the best performance, followed by SW+SVM{1LDS} [30]. Particularly, the introduction of timing structure in the BoS codebook generation allows the model to overcome possible confusions when dealing with $N > 1$ LDS. This can explain the better performances against SW+5NN{NLDS} and SW+SVM{NLDS}. We ran the tests several times and found the results consistent.

Other breakdance sequences from [34] show similar performance as Free. Although the performances are different, the surface dynamics of rigid and non-rigid regions show same characteristics.

As well, we computed the confusion matrices for classes from different sequences. In Tables 1 and 2, we show results for Free and Samba sequences using SW+SVM{1LDS} [30] and SWTF+SVM{NLDS} as they were the best performers in the previous experiments. Here, all surface patches from all the test datasets from the Free sequence were tested for classification using patches from the Samba sequence as training data. Our approach outperforms [30]. This is primarily because dynamic surfaces for performance capture can exhibit various behaviors in time that cannot be well modeled using a single LDS per feature. Besides, confusions in patch classification can be due to the bottom of the dress of the Samba dancer, as the transitions between the tight and loose parts and unclear, even for manual classification. On the other hand, SWTF+SVM{NLDS} did no mistake when classifying non-rigid region patches.

Table 1. Confusion matrix of SW+SVM{1LDS} [30].

	Samba rigid	Samba non-rigid
Free rigid	50%	50%
Free non-rigid	75%	25%

Table 2. Confusion matrix of SWTF+SVM{NLDS}[ours].

	Samba rigid	Samba non-rigid
Free rigid	87.5%	12.5%
Free non-rigid	0	100%

Dynamic surface segmentation. Surface region characterization allows body part segmentation, as skin and clothing can potentially be identified using surface patch dynamics. However this is a challenging task when no prior on the object is given, and it is still an active research field [16].

7. Conclusion

As 3D reconstruction technologies have become capable to capture surface deformations in real-time and details such as clothing wrinkles, dynamic surfaces representing human performance can now be characterized using local geometry information. Moreover, assuming dynamic surfaces as streams of temporally continuous and indefinitely varying data having certain temporal statistics, we can draw an analogy with the dynamic textures. Hence in this paper, we present the following contributions: 1) no prior work has addressed dynamic surface characterization using surface intrinsic properties (such as local curvatures), 2) we propose to model surface dynamics using hybrid linear dynamical system models (i.e., with N LDS per surface feature) within the bag of dynamical systems (BoS) framework, and 3) we introduce LDS timing structure in the codebook formation of the BoS. We show experimental results on datasets of real-world dynamical surfaces for description, classification and segmentation. As well, we discuss the accuracy and quality at small scale of existing datasets of dynamic surfaces. Other datasets to consider are soft tissue organs (e.g., heart, lungs) for anomaly or disease detection, and fluids. We believe our model has great potential for future research and applications as 3D sensing technologies are rapidly becoming even more accurate.

Acknowledgements

The authors would like to thank Dr. Hiroaki Kawashima from Kyoto University for the precious discussions about dynamical system modeling. This work was supported by the JST-CREST project "Creation of Human-Harmonized Information Technology for Convivial Society".

Appendix: Surface patches

Figure 6 shows surface patches for the models from the sequences Free, Samba and Bouncing, computed as in [4]. We show as well mapping of patches on square parametrization domain, obtained using the same transformations as in Fig. 1 and Fig. 3, as it is a good representation to understand the analogy with dynamic textures. However, we recall that curvature maps are used only for visualization and assessment of surface point tracking, and not for the tracking itself (as surfaces are cut and geometry is distorted).

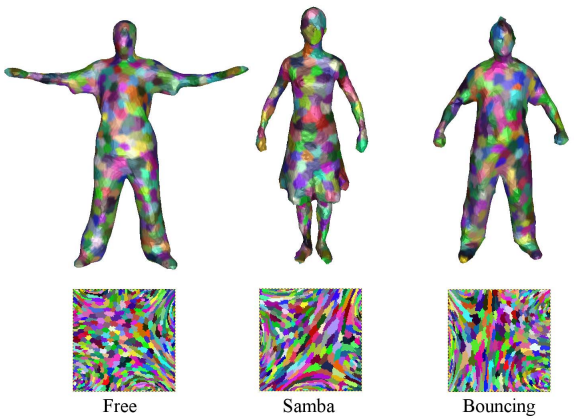

Figure 6. Surface patches for the models from the sequences Free, Samba and Bouncing and projections on square domain given for visualization and assessment of surface point tracking.

References

[1] B. Afsari, R. Chaudhry, A. Ravichandran, and R. Vidal. Group action induced distances for averaging and clustering

linear dynamical systems with applications to the analysis of dynamic scenes. *CVPR*, 2012. 4

[2] J. Allard, C. Ménier, B. Raffin, E. Boyer, and F. Faure. Grimage: Markerless 3d interactions. *SIGGRAPH - Emerging Technologies*, 2007. 2, 5, 6

[3] M. Bojsen-Hansen, H. Li, and C. Wojtan. Tracking surfaces with evolving topology. *SIGGRAPH*, 2012. 2

[4] C. Cagniart, E. Boyer, and S. Ilic. Probabilistic deformable surface tracking from multiple videos. *ECCV*, 2010. 2, 3, 4, 6, 7

[5] H. Cetingul and R. Vidal. Intrinsic mean shift for clustering on stiefel and grassmann manifolds. *CVPR*, 2009. 4

[6] A. B. Chan, E. Coviello, and G. R. G. Lanckriet. Clustering dynamic textures with the hierarchical em algorithm. *CVPR*, 2010. 4

[7] A. B. Chan and N. Vasconcelos. Mixtures of dynamic textures. *ICCV*, 2005. 1, 2, 4

[8] A. B. Chan and N. Vasconcelos. Classifying video with kernel dynamic textures. *CVPR*, 2007. 6

[9] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal. Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. *CVPR*, 2009. 2, 3

[10] K. M. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette across time: Part ii: Applications to human modeling and markerless motion tracking. *IJCV*, 63(3):225–245, 2005. 2

[11] E. de Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel, and S. Thrun. Performance capture from sparse multi-view video. *ACM Trans. Graphics*, 27(3), 2008. 1, 2, 4, 5, 6

[12] E. de Aguiar, C. Theobalt, C. Stoll, and H.-P. Seidel. Markerless deformable mesh tracking for human shape and motion capture. *CVPR*, 2007. 2

[13] G. Doretto, A. Chiuso, Y. Wu, and S. Soatto. Dynamic textures. *IJCV*, 51(2):91–109, 2003. 1, 2, 3, 4

[14] G. Doretto, D. Cremers, P. Favaro, and S. Soatto. Dynamic texture segmentation. *ICCV*, 2003. 2

[15] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral grouping using the nystrom method. *PAMI*, 2009. 5

[16] J. Franco and E. Boyer. Learning temporally consistent rigidities. *CVPR*, 2011. 2, 7

[17] J. Franco, C. Menier, E. Boyer, and B. Raffin. A distributed approach for real-time 3d modeling. *CVPR Workshop on Real-Time 3D Sensors and their Applications*, page 31, 2004. 1, 2

[18] Y. Furukawa and J. Ponce. Dense 3d motion capture from synchronized video streams. *CVPR*, 2008. 2

[19] P. Huang, A. Hilton, and J. Starck. Shape similarity for 3d video sequences of people. *IJCV Special Issue on 3D Object Retrieval*, 89(2-3):362–381, 2010. 2

[20] H. Jiang, H. Liu, P. Tan, G. Zhang, and H. Bao. 3d reconstruction of dynamic scenes with multiple handheld cameras. *ECCV*, 2012. 1, 2

[21] Y.-G. Jiang, C.-H. Ngo, and J. Yang. Towards optimal bag-of-features for object categorization and semantic video retrieval. *CIVR*, 2007. 4, 5

[22] T. Kanade, A. Yoshida, K. Oda, H. Kano, and M. Tanaka. A stereo machine for video-rate dense depth mapping and its new applications. *CVPR*, 1996. 2

[23] L. Kaufman and P. Rousseeuw. Clustering by means of medoids. *Statistical Data Analysis Based on the L1-Norm and Related Methods, Y. Dodge Ed., North-Holland*, 1987. 4, 5

[24] H. Kawashima and T. Matsuyama. Interval-based modeling of human communication dynamics via hybrid dynamical systems. *NIPS Workshop on Modeling Human Communication Dynamics*, 2010. 2, 4

[25] J. Koenderink and A. van Doorn. Surface shape and curvature scales. *Image and Vision Computing*, 1992. 2

[26] Z. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *CVPR*, 2006. 4

[27] T. Matsuyama, X. Wu, T. Takai, and S. Nobuhara. Real-time 3d shape reconstruction, dynamic 3d mesh deformation, and high fidelity visualization for 3d video. *CVIU*, 96(3):393–434, 2004. 1, 2

[28] M. Ovsjanikov, Q. Merigot, F. Memoli, and L. J. Guibas. One point isometric matching with the heat kernel. *Comput. Graph. Forum*, 29(5):1555–1564, 2010. 2

[29] L. R. Rabiner. A tutorial on hidden markow models and selected applications in speech recognition. *IEEE*, 77(2):257–286, 1989. 2

[30] A. Ravichandran, R. Chaudhry, and R. Vidal. View-invariant dynamic texture recognition using a bag of dynamical systems. *CVPR*, 2009. 1, 2, 3, 4, 5, 6, 7

[31] L. Saboret, P. Alliez, and B. Lévy. Planar parameterization of triangulated surface meshes. *In CGAL Reference Manual. CGAL Editorial Board, 4.0 edition*, 2012. 3

[32] P. Saisan, G. Doretto, Y. Wu, and S. Soatto. Dynamic texture recognition. *CVPR*, 2001. 2, 3, 4, 6

[33] J. Starck and A. Hilton. Spherical matching for temporal correspondence of non-rigid surfaces. *ICCV*, 2005. 2

[34] J. Starck and A. Hilton. Surface capture for performance-based animation. *IEEE Computer Graphics and Applications*, 2007. 1, 2, 3, 5, 6, 7

[35] T. Tung and T.Matsuyama. Dynamic surface matching by geodesic mapping for 3d animation transfer. *CVPR*, 2010. 2

[36] T. Tung and T.Matsuyama. Invariant surface-based shape descriptor for dynamic surface encoding. *ACCV*, 2012. 3

[37] T. Tung and T.Matsuyama. Topology dictionary for 3d video understanding. *PAMI*, 34(8):1645–1657, 2012. 2, 5

[38] K. Varanasi, A. Zaharescu, E. Boyer, and R. Horaud. Temporal surface tracking using mesh evolution. *ECCV*, 2008. 2

[39] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. *PAMI*, 27(1):475– 480, 2005. 2

[40] R. Vidal and A. Ravichandran. Optical flow estimation and segmentation of multiple moving dynamical textures. *CVPR*, 2005. 2

[41] A. Zaharescu, E. Boyer, K. Varanasi, and R. Horaud. Surface feature detection and description with applications to mesh matching. *CVPR*, 2009. 2