Towards Plenoptic Dynamic Textures

Gianfranco Doretto
UCLA Computer Science Department
Los Angeles, CA 90095
Email: doretto@cs.ucla.edu

Stefano Soatto
UCLA Computer Science Department
Los Angeles, CA 90095
Email: soatto@ucla.edu

***Abstract*— We present a technique to infer a model of the spatio-temporal statistics of a collection of images of dynamic scenes seen from a moving camera. We use a time-variant linear dynamical system to jointly model the statistics of the video signal and the moving vantage point. We propose three approaches to inference, the first based on the plenoptic function, the second based on interpolating linear dynamical models, the third based on approximating the scene as piecewise planar. For the last two approaches, we also illustrate the potential of the proposed techniques with a number of experiments. The resulting algorithms could be useful for video editing where the motion of the vantage point can be controlled interactively, as well as to perform stabilized synthetic generation of video sequences.**

I. Introduction

Images of objects with complex shape, motion and material properties are commonplace in our visual world: think of a silk gown, a burning flame, a waterfall. The complexity of these physical phenomena is far superior to that of the images that they generate and, therefore, the inverse problem of visual perception is intrinsically ill-posed. For instance, the rivulets on the surface of a lake could be the result of homogeneous material (water) being tossed around by the wind and the currents; however, the same perception could be elicited by a giant flat screen where a time-varying signal is projected to match the appearance of the rivulets, adapted to the viewer's moving vantage point. Therefore, unless additional prior assumptions are available (as usually done in "shape from X" algorithms), one has to give up the goal of inferring "*the*" model of the physical world, and settle for a much poorer representation, one that can explain the measured data. What representation to choose depends on the task at hand.

In this paper, we are interested in inferring models of the spatio-temporal statistical properties of visual scenes that can be used to generate synthetic sequences of images, where both the temporal statistics and the motion of the vantage point can be edited.

A. Related work and contributions of this paper

Our goal is simple to state: we are given images of a dynamic scene that exhibit some sort of spatio-temporal regularity, taken from a moving camera, and we want to extract a model so that we can re-play the sequence from an arbitrary vantage point. As simple as the goal sounds, the discussion above suggests that it is unattainable. Since we know at the outset that we cannot retrieve a physically correct model of the shape, dynamics and material properties of the scene, we will let our task guide our assumptions on the representation to lead to a well-posed inference problem.

We will model the images (or filtered versions of the images) as the output of a time-varying dynamical model. The model, together with a stochastic input, represents the dynamic variability of the image sequence. In addition, we explicitly model the vantage point, so that during the synthesis, we can change it arbitrarily and render sequences from a camera undergoing a virtual motion.

Several algorithms have been proposed for interpolating and extrapolating views of a scene without explicit reconstruction (see for instance [1], [2] and references therein), as well as techniques to model changes in the viewpoint that use little or no scene structure information (e.g. [3], [4] and also [5], [6], that use the Bidirectional Texture Function [7]). All of these techniques, however, require the scene to be static (or a large number of static calibrated cameras), whereas we are interested specifically in modeling dynamic scenes.

The literature on modeling dynamic scenes is also sizeable (e.g. [8], [9] and references therein). Most of these techniques, however, consider scenes made of a number of rigid objects, whereas we are interested in scenes where the temporal dynamics of the entire scene can be modeled, and no rigidity constraints are available.

Image-based rendering techniques are available for specific classes of non-rigid objects, (e.g. for facial expression animations [10], [11]). Here instead, we are interested in scenes for which no prior information is available, and allow complex shape, motion, and material properties. We will, however, make assumptions on the "temporal regularity" of the scene, in ways that we will explain in later sections.

This latter class of scenes has recently received some attention, and there are techniques to model changes in the dynamics of such scenes (e.g. [12], [13], [14], [15] end references therein). Therefore, one can find techniques to model only changes in the viewpoint, as mentioned above, or only changes in the dynamics of the scene.

The first attempt to model changes in both vantage point and dynamics of the scene is, to the best of our knowledge, the work of Fitzgibbon [13]. There, the author was looking for sequence registration for nowhere static scenes where the decomposition into parametric camera motion and stochastic motion of the scene is still possible.

Of the three approaches we suggest in Sect. III, [13] is most closely related to the third, which considers a homographic approximation of the scene, although with a different purpose (matching and registration for [13], modeling for synthesis in our case). Even so, we use an alternating minimization where each step is guaranteed to reduce the global cost, which is

different from [13]. In addition, we propose and experiment with direct dynamical model interpolation. Also, we propose modeling the dynamics of the plenoptic function [16] directly.

In the next section we introduce the formalism to be used throughout the paper, and in Section III we propose three approaches to model moving images of dynamic scenes for the purpose of view synthesis and rendering. The performance of two of these approaches are explored in the experimental Section IV.

II. Formalization of the problem

Let $\mathbf{X} \in \mathbb{R}^3$, and $I(\mathbf{X}, t) \in \mathbb{R}_+$ be the "energy" of a particle in position $\mathbf{X}$ in space at time t. An image $y(\mathbf{x}, t)$ at a pixel $\mathbf{x} \in \Omega \subset \mathbb{R}^2$ and at time t, is in general obtained by integrating the energy along the projection ray $\mathbf{x}\lambda \in \mathbb{R}^3$, where $\lambda \in \mathbb{R}_+$ and $\mathbf{x}$ is expressed in homogeneous (projective) coordinates: $y(\mathbf{x}, t) = \int_{\mathbb{R}_+} I(\mathbf{x}\lambda, t) d\lambda$. If we assume that particles are opaque, or that they are concentrated on a surface, then the integral reduces to the value of I at the particular $\mathbf{X}$ that corresponds to $\mathbf{x} = \pi(\mathbf{X})$, where π is the canonical projection: if $\mathbf{X} = \mathbf{x}\lambda$, then $\pi(\mathbf{X}) = \mathbf{x}$. More in general, let $P(\mathbf{x}) \in \mathbb{R}^3$ be the surface in space that contributes to the image irradiance y at $\mathbf{x}$, and let the viewpoint move according to a motion described by a group $g(t)$ (Euclidean, affine or projective). Therefore, under these assumptions, we have $y(\mathbf{x}, t) = I(g(t)P(\mathbf{x}), t) + w(\mathbf{x}, t)$ where w is a measurement noise term that we assume to be white and zero-mean. Now, if we further allow the surface P to change over time, we have the image formation model in the most general form that we will address in this paper:

$$y(\mathbf{x}, t) = I\left(g(t)P(\mathbf{x}, t), t\right) + w(\mathbf{x}, t). \tag{1}$$

The goal is, given measurements of $y(\mathbf{x}, t)$ for $\mathbf{x} \in \Omega \subset \mathbb{R}^2$ and $t = 1, \ldots, \tau$, to recover a model of the form above, consisting of the unknowns $I(\cdot, \cdot), g(\cdot)$ and $P(\cdot, \cdot)$, such that novel sequences can be generated by altering the model or controlling its states.

A. Reduction of the model

Unfortunately, the model (1) is "too rich," in that given any measured sequence $\{y(\mathbf{x}, t),\ \mathbf{x} \in \Omega, 0 \leq t \leq \tau\}$ there exist infinitely many models I, g, P that generate them. Therefore, learning would be subject to overfitting and the resulting model would have little predictive power. In fact, if we write the model (1) in more compact form as $I_t \circ g \circ P_t$, then it is immediate to see that, for any choice $\tilde{g} \in SE(3)$ and arbitrary $\tilde{P}(\mathbf{x}, t)$, we can always choose $\tilde{I}(\mathbf{X}, t)$ that satisfies the equation $I_t \circ g \circ P_t = \tilde{I}_t \circ \tilde{g} \circ \tilde{P}_t$. Therefore, we need to restrict the class of allowable energy fields I. In the following, we will assume that I are subject to a temporal dynamics that is second-order stationary. That is, $I(\mathbf{X}, t)$ is *allowable* if there exist $C(\mathbf{X})$, A and $\xi(t)$ such that

$$\begin{cases} \xi(t+1) = A\xi(t) + v(t) \quad \xi(0) = \xi_0\ ; \quad v \sim \mathcal{N}(0, Q) \\ y(\mathbf{x}, t) = C(\mathbf{X})\xi(t) + w(\mathbf{x}, t) \end{cases} \tag{2}$$

for some v, w white, zero-mean Gaussian random processes, and $I(\mathbf{X}, t) = C(\mathbf{X})\xi(t)$. In other words, $I(\cdot, t)$ is a 3D linear dynamic texture [14]. Notice that $I(\mathbf{X}, t)$ cannot be measured for all $\mathbf{X} \in \mathbb{R}^3$, but it is instead sampled on a set of measure zero determined by $\mathbf{X}(t) = g(t)P(\mathbf{x}, t)$.

As restrictive as this model may appear, it is not enough to guarantee a one-to-one correspondence between parameters and output realizations. In fact, given C, g, P and ξ, one can always find $\tilde{C}, \tilde{g}, \tilde{P}, \tilde{\xi}$ that satisfy $\tilde{C}(\tilde{g}(t)\tilde{P}(\mathbf{x}, t))\tilde{\xi}(t) = C(g(t)P(\mathbf{x}, t))\xi(t)$. Indeed, one can choose a function $\tilde{C}(\cdot)$ and $\tilde{g}(t)$ arbitrarily, and always find $P(\mathbf{x}, t)$ that satisfies the equation above. Therefore, we need to restrict P. One possibility is to assume that P is a static surface, and only the viewpoint $g(t)$ and the radiance $I(\cdot, t)$ are allowed to change over time. In that case, we have

$$I(\mathbf{X}, t) = C(g(t)P(\mathbf{x}))\xi(t)\ . \tag{3}$$

III. Three approaches

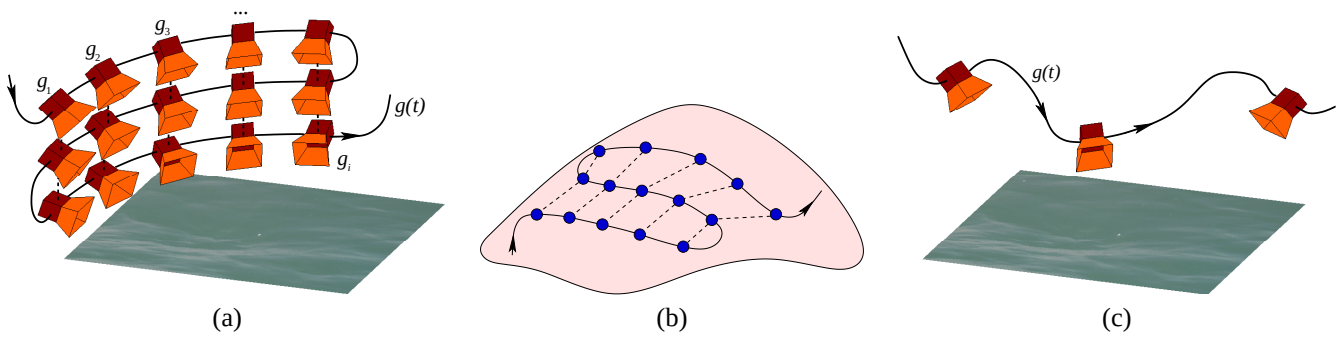
In the next three subsections we propose three approaches to learn and synthesize dynamic textures as seen from a moving camera. In Section III-A we propose an operative model that is conceptually straightforward but difficult to implement in practice, because it requires a large number of calibrated and synchronized cameras. In Section III-B we propose an interpolation technique to interpolate time-invariant instances of the general model proposed in Section III-A, and does not require calibration. Finally, in Section III-C we propose a further reduction of model (3) in which we assume that P is not only a static surface, but can be approximated locally by a plane.

A. The Dynamic Lumigraph

The first and conceptually simplest approach to modeling and learning dynamic textures as seen from a moving vantage point is to start directly from the model (2), and collect data for a large set of "voxels" $\mathbf{X}$ and viewpoints g. At that point, synthesis is trivially performed by choosing $g(t)$ and P via

$$y_{synth}(\mathbf{x}, t; g(t), P(\mathbf{x}, t)) = C(g(t)P(\mathbf{x}, t))\xi(t). \tag{4}$$

This approach is equivalent to a dynamic version of the so-called Lumigraph [3], and aims at modeling the plenoptic function [16] directly, this is why we refer to sequences modeled by (2) and (4) as *plenoptic dynamic textures*. Although conceptually viable, this approach is impractical because it would require a large number of synchronized cameras, one per desired location g. In the Lumigraph, the camera is moved, so that time is used to sample space, a trick that we cannot apply here since we need to sample both in space and time as our scene is not static. Since we do not have an experimental facility that would allow us to collect synchronized images, this avenue is not pursued in the experimental Section IV. Another research group is currently building a rig of 128 calibrated and synchronized cameras, so testing this approach will be feasible within the next few years.



(a) (b) (c)

Fig. 1. (a) interpolation of the camera trajectory in $SE(3)$, requires knowledge of the extrinsic calibration, which can be obtained with a laborious procedure. (b) interpolation in the space of model parameters can be easily attained by building the connectivity graph by computing mutual distances among each model. (c) under the homographic approximation, one can estimate model parameters along a one-dimensional subset of camera trajectories.

B. Model interpolation

Consider a camera trajectory, represented by $g(t) \in SE(3)$. This is a one-dimensional subset of the six-dimensional space of all possible camera poses, and we are interested in interpolating and extrapolating the viewpoint so as to be able to move the camera arbitrarily in space.

If enough viewpoints $g(t)$ are sampled uniformly in $SE(3)$, then one could think of generating synthetic sequences by interpolating models of the form (2) from neighboring g's. Naturally, this *requires knowing the pose of the camera* that captured the original data, i.e. that the (extrinsic) calibration of the camera be known. Most often, however, the camera pose is not known precisely, which is another shortcoming of the straightforward extension of the Lumigraph to dynamic objects.

Consider now the set of models corresponding to a trajectory of viewpoint samples $g_1, g_2, \ldots, g_i$ (see Figure 1-(a)). These are points in the space of linear-Gaussian models (see Figure 1-(b)), which can be endowed with a metric structure, that allows determining the complete distance graph between each sample model. Therefore, one can envision generating synthetic sequences by generating local interpolations in the space of models. The shortcoming of this method is that one cannot manipulate the viewpoint directly, and therefore the editing power is reduced. Interpolation can be performed once the connectivity graph is available. In order to compute it, one needs to define a norm in the space of models. This can be done in a variety of ways. We choose subspace angles among extended observability subspaces, as proposed by De Cock and De Moor; the interested reader can refer to [17] for details.

In order to compute the interpolation between two models, we need to compute geodesics in the space of models. As pointed out in [14], the matrices A, C, and Q, are supposed to have a certain structure. In particular, C must have orthonormal columns, A must have the eigenvalues within the unit circle of the complex plane, and Q must be symmetric positive definite. Computing geodesics in such a manifold is an open problem, and in Section IV we show some results attained using a sub-optimal technique (which uses rotations and SVDs), that still enforces the properties of model parameters.

C. Piecewise planar approximation

In this section we suppose that the scene can be roughly approximated by a plane, so that $g(t)P(\mathbf{x})$ can be represented as a homography of the image plane[1] $H(t) \in \mathbb{R}^{3\times 3}/\mathbb{R}$, and therefore we have $I(\mathbf{X}, t) = C(H(t)\mathbf{x})\xi(t)$, where $\mathbf{x}$ is expressed in homogeneous coordinates. We also need a dynamical model for the evolution of $H(t)$: in lack of a better model we will assume it to be a first-order random walk. Summarizing this simplified model, we have

$$\begin{cases} \xi(t+1) = A\xi(t) + v(t) \quad \xi(0) = \xi_0 \ ; \quad v \sim \mathcal{N}(0, Q) \\ H(t+1) = H(t) + v_H(t) \quad H(0) = I \\ y(\mathbf{x}, t) = C(H(t)\mathbf{x})\xi(t) + w(\mathbf{x}, t) \ . \end{cases} \tag{5}$$

Now, given $\{y(\mathbf{x}, t), \ \mathbf{x} \in \Omega, t = 1, \ldots, \tau\}$, the problem of inferring a model consists of estimating $A, C(\cdot), H(t), \xi(t)$ as well as the covariance of the noise terms v, v_H and w. The matrix C is inferred only along the one-dimensional subset of camera trajectories, and therefore synthesis will be limited to a (homographic) neighborhood of this trajectory[2]. This setup is illustrated in Figure 1-(c).

The pedestrian way to infer the state of model (5) consists of first estimating the homography off-line, for instance by registering a set of point features that are known to be static, and then applying any of the modeling algorithms as if the viewpoint was fixed[3], for instance [14], [15].

A more convenient and practical algorithm, which we explore in Section IV, consists of setting up an alternating minimization problem where we start with $\hat{H}_0 = I$, $\hat{\xi}_0 = 0$, and at a generic iteration i, we alternate a step of the subspace identification algorithm N4SID (see [18] for details)

$$\hat{A}_{i+1}, \hat{C}_{i+1}, \hat{Q}_{i+1} = \text{N4SID}(\{y(\hat{H}_i^{-1}\mathbf{x}, t)\}_{t=1,\ldots,\tau}) \ , \tag{6}$$

[1] If the plane has normal vector $\nu \in \mathbb{R}^3$ relative to the camera frame, and moves with a Euclidean motion $g(t) \in SE(3)$, represented by a rotation matrix $R(t) \in SO(3)$ and a translation vector $T(t) \in \mathbb{R}^3$, then $H(t) = R(t) + T(t)\nu^T$ up to a scale factor.

[2] Naturally, if the scene is planar, this neighborhood spans the entire space, and therefore one can generate views from an arbitrary vantage point

[3] This corresponds to defining $\tilde{y}(\mathbf{x}, t) \doteq y(H^{-1}\mathbf{x}, t) = y(\tilde{\mathbf{x}}(t), t) = C(\tilde{\mathbf{x}}(t), t)\xi(t) + w(\tilde{\mathbf{x}}(t), t)$ where, by assumption, $C(\tilde{\mathbf{x}}(t), t) = C(\tilde{\mathbf{x}}(0), 0)$ is constant.

with a step along the gradient of the cost function $\phi_i \doteq \sum_t \|y(H_i^{-1}\mathbf{x},t) - \hat{C}_{i+1}\hat{\xi}(t)\|^2$, subject to $\hat{\xi}(t+1) = \hat{A}_{i+1}\hat{\xi}(t)$. This results in

$$\hat{H}_{i+1}^{-1}(t) = \hat{H}_i^{-1}(t) + \alpha_i \frac{\partial \phi_i}{\partial H_i^{-1}(t)}(\hat{H}_i^{-1}(t)) \ , \qquad (7)$$

where[4] $\frac{\partial \phi_i}{\partial H_i^{-1}(t)}(\hat{H}_i^{-1}(t))$ is given by the expression

$$2\sum_{\mathbf{x}\in\Omega}(y(\hat{H}_i^{-1}\mathbf{x},t) - \hat{C}_{i+1}\hat{\xi}(t))\nabla y(\hat{H}_i^{-1}\mathbf{x},t)D(\hat{H}_i^{-1}\mathbf{x},t) \ , \qquad (8)$$

in which, if we call $\mathbf{x}(t) = H^{-1}(t)\mathbf{x}$, $D(\mathbf{x},t)$ is given by

$$\frac{1}{x_3(t)}\begin{bmatrix} \mathbf{x}^T & 0 & -\frac{x_1(t)}{x_3(t)}[x_1,\ x_2] \\ 0 & \mathbf{x}^T & -\frac{x_1(t)}{x_3(t)}[x_1,\ x_2] \end{bmatrix} \in \mathbb{R}^{2\times 8}. \qquad (9)$$

Once the model is identified, and $\hat{A}$, $\hat{C}$, and $\hat{Q}$ are given, one can easily generate novel sequences by choosing an arbitrary camera path $\{(R(t),T(t))\}_{t=1,2,\ldots}$, sampling an input noise $v \sim \mathcal{N}(0,\hat{Q})$, and generating $H(t) = R(t) + T(t)\nu^T$. The sequence of images $y_{synth}(t)$ is then produced by iterating the model (5) forward in time starting from an arbitrary initial condition[5].

IV. Experiments

We tested the approach described in Section III-B on two data sets that we call *inverted-fountain* and *waterfall*. The first one consists of 6 sequences of 120 color frames of 350×240 pixels. From the first to the last sequence the camera is approximately sampling a circular trajectory that pans around the inverted-fountain, see Figure 2-(a) for a sample of the data set. The second data set consists of 21 sequences of 120 color frames of 320×240 pixels, that almost uniformly sample a portion of the 3D space.

For the inverted-fountain data set we do synthesis by concatenating forward and backward the 6 models inferred by the 6 sequences. The resulting movie appears to be made by a camera that is panning smoothly around the inverted-fountain on a circular trajectory, as we would have expected to see. The movie is 240 frames long.

For the waterfall data set we compute the connectivity graph and we extract 6 models along a path that goes through 3 key-models. The key-models were selected manually while the other three were selected automatically by minimizing the path in the connectivity graph. From the first key-model to the second one the camera is moving closer to the scene; from the second to the third key-model the camera is panning around the scene; from the third key-model we simply go back to the second. The resulting movie appears to be made by a camera that is moving forward and then panning around the scene back and forth. The movie is 200 frames long.

We tested the procedure described in Section III-C with two sequences that we call *fountain-corner* and *waterfall-2*. The former has 170, and the latter 130 color frames of 350×240 pixels. The state dimension for all the models we used was set to 50. The rows (e) and (g) of Figure 2 show samples of the original sequences, and the same samples after the rectification with respect to the estimated homographies. The rows (f) and (h) show some synthesized frames. The synthesized movies are 200 frames long, and the frame dimension is 175×120 pixels.

Notice that the piecewise planar technique can also be used to stabilize scenes with complex dynamics. For instance, the original movie waterfall-2 is very jittery (due to the hand-held camera), while during the synthesis one can generate arbitrary smooth motions, and preserve the dynamic appearance.

V. Conclusions

We have presented three approaches to model the spatio-temporal statistics of a collection of images as seen from a moving camera. For two of them we proposed algorithms for identifying the model and perform synthesis of plenoptic dynamic textures. Although the model does not capture the physics of the scene, it is sufficient to "explain" the measured data and extrapolate the appearance of the images in space and time. Unlike the model interpolation approach, the piecewise planar approximation allows full editing power in terms of controlling the motion of the vantage point. This technique could also be used to stabilize scenes with complex dynamics.

Acknowledgments

This research was supported by the following grants: NSF ECS-0200511, CCR-0121778, AFOSR F49620-03-1-0095.

References

[1] S. Avidan and A. Shashua, "Novel view synthesis in sensor space," in *Proc. Computer Vision and Pattern Recognition Conf.*, 1997, pp. 1034–1040.

[2] S. M. Seitz and C. R. Dyer, "View morphing," in *Proc. SIGGRAPH '96*, 1996, pp. 21–30.

[3] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Choen, "The lumigraph," in *Proc. SIGGRAPH '96*, 1996, pp. 43–54.

[4] M. Levoy and P. Hanrahan, "Light field rendering," in *Proc. SIGGRAPH '96*, 1996, pp. 161–170.

[5] X. Liu, Y. Yu, and H. Y. Shum, "Synthesizing bidirectional texture functions for real-world surfaces," in *Proc. SIGGRAPH '01*, 2001, pp. 97–106.

[6] A. Zalesny and L. V. Gool, "Multiview texture models," in *Proc. Conf. on Computer Vision and Pattern Recognition*, vol. 1, 2001, pp. 615–622.

[7] K. J. Dana, B. van Ginneken, S. K. Nayar, and J. J. Koenderink, "Reflectance and texture of real-world surfaces," *ACM Transactions on Graphics*, vol. 18, no. 1, pp. 1–34, 1999.

[8] R. A. Manning and C. R. Dyer, "Interpolating view and scene motion by dynamic view morphing," in *Proc. Computer Vision and Pattern Recognition Conf.*, vol. 1, 1999, pp. 388–384.

[9] M. Irani, P. Anandan, and S. Hsu, "Mosaic based representations of video sequences and their applications," in *Proc. 5th Int. Conf. on Computer Vision*, 1995, pp. 605–611.

[4] Equation (7) entails a slight abuse of notation, since it applies to H represented as a $9-$dimensional vector (rather than a 3×3 matrix), with the component h_{33} set to one. This technique is related to the work of Fitzgibbon [13], although the use of N4SID simplifies the optimization task and guarantees (local) convergence due to the optimality and asymptotic efficiency of the algorithm [19].

[5] Indeed, by allowing more general changes in $H(t)$ one can simulate changes of the internal parameters of the camera and simulate changes in focal length (zoom), aperture (field of view) etc. Since the synthesis phase is computationally trivial, the viewpoint can be manipulated interactively (in real time), for instance from an input device with six degrees of freedom, such as a joystick.

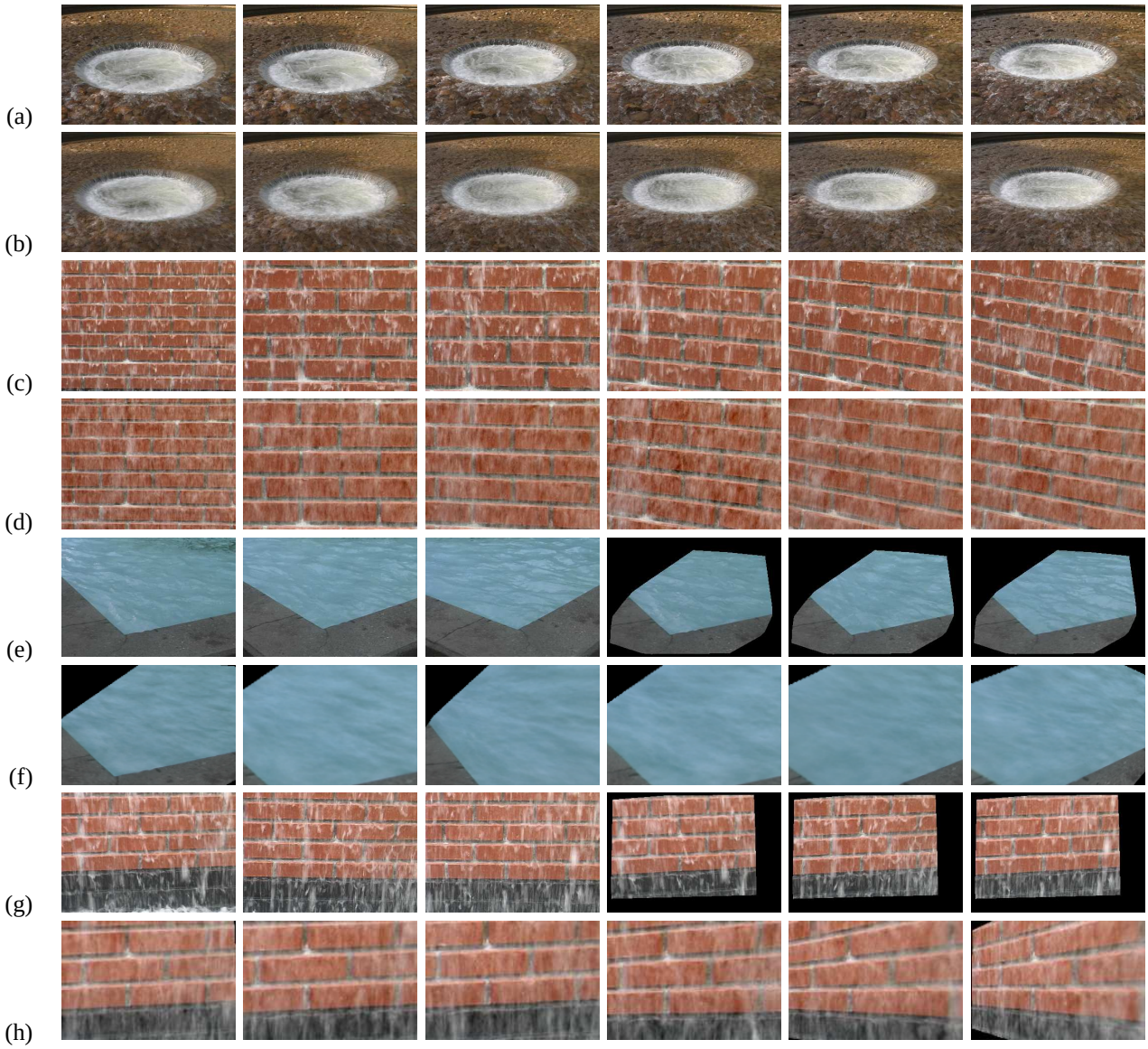


Fig. 2. **Inverted-fountain.** (a) samples of the 6 original sequences of the data set. The positions $g_1, \ldots, g_6$ are such that the camera is sampling a rotation trajectory around the fountain. (b) samples of a synthesized sequence. **Waterfall.** (c) samples of the 6 original sequences of the data set. The positions $g_1, \ldots, g_6$ are such that the camera is sampling a trajectory that first move forward and than rotates around the waterfall. (d) samples of a synthesized sequence. The frames in rows (b) and (d) are obtained using the procedure described in Section III-B. **Fountain-corner.** (e) three samples of the original sequence, and their corresponding samples after the homography registration. (f) samples of a synthesized sequence. The synthesized camera motion is such that the camera is first zooming in, then translating to the left, turning to the left, right, and finally zooming out. **Waterfall-2.** (g) three samples of the original sequence, and their corresponding samples after the homography registration. (h) samples of a synthesized sequence. The synthesized camera motion is such that the camera is first zooming in, than translating to the left, down, right, up, and finally rotating to the left. The frames in rows (f) and (h) are obtained using the procedure described in Section III-C. These results are best seen in the movies available on-line [20].

[10] C. Bregler, M. Covell, and M. Slaney, "Video rewrite: driving visual speech with audio," in *Proc. SIGGRAPH '97*, 1997, pp. 353–360.

[11] G. J. Edwards, C. J. Taylor, and T. F. Cootes, "Learning to identify and track faces in image sequences," in *Proc. 6th Int. Conf. on Computer Vision*, 1998, pp. 317–322.

[12] A. Schödl, R. Szeliski, D. H. Salesin, and I. Essa, "Video textures," in *Proc. SIGGRAPH '00*, July 2000, pp. 489–498.

[13] A. W. Fitzgibbon, "Stochastic rigidity: image registration for nowhere-static scenes," in *Proc. Int. Conf. on Computer Vision*, vol. 1, July 2001, pp. 662–669.

[14] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto, "Dynamic textures," *Int. Journal of Computer Vision*, vol. 51, no. 2, pp. 91–109, February 2003.

[15] Y. Z. Wang and S. C. Zhu, "A generative method for textured motion analysis and synthesis," in *Proc. European Conf. on Computer Vision*, vol. 1, June 2002, pp. 583–597.

[16] E. H. Adelson and J. R. Bergen, "The plenoptic function and the elements of early vision," in *Computational Models of Visual Processing*. MIT Press, Cambridge, MA, 1991, pp. 3–20.

[17] K. D. Cock and B. D. Moor, "Subspace angles between linear stochastic models," in *Proc. 39th Int. Conf. on Decision and Control*, vol. 2, Dec 2000, pp. 1561–1566.

[18] P. V. Overschee and B. D. Moor, "N4sid: subspace algorithms for the identification of combined deterministic-stochastic systems," *Automatica*, vol. 30, pp. 75–93, Jan 1994.

[19] D. Bauer, M. Deistler, and W. Scherrer, "Consistency and asymptotic normality of some subspace algorithms for systems without observed inputs," *Automatica*, vol. 35, no. 7, pp. 1243–54, July 1999.

[20] http://www.cs.ucla.edu/∼doretto/projects/viewpoint-editing.html.