Stochastic rigidity: Image registration for nowhere-static scenes.

Andrew W. Fitzgibbon
Robotics Research Group, Department of Engineering Science,
University of Oxford, 19 Parks Road, Oxford OX1 3PJ, United Kingdom

Abstract

We consider the registration of sequences of images where the observed scene is entirely non-rigid; for example a camera flying over water, a panning shot of a field of sunflowers in the wind, or footage of a crowd applauding at a sports event. In these cases, it is not possible to impose the constraint that world points have similar colour in successive views, so existing registration techniques [1, 5, 9, 11] cannot be applied. Indeed the relationship between a point's colours in successive frames is essentially a random process.

However, by treating the sequence of images as a set of samples from a multidimensional stochastic time-series, we can learn a stochastic model (e.g. an AR model [16, 23]) of the random process which generated the sequence of images. With a static camera, this stochastic model can be used to extend the sequence arbitrarily in time: driving the model with random noise results in an infinitely varying sequence of images which always looks like the short input sequence. In this way, we can create "videotextures" [21, 24] which can play forever without repetition.

With a moving camera, the image generation process comprises two components—a stochastic component generated by the videotexture, and a parametric component due to the camera motion. For example, a camera rotation induces a relationship between successive images which is modelled by a 4-point perspective transformation, or homography. Human observers can easily separate the camera motion from the stochastic element.

The key observation for an automatic implementation is that without image registration, the time-series analysis must work harder to model the combined stochastic and parametric image generation. Specifically, the learned model will require more components, or more coefficients, to achieve the same expressive power as for the static scene. With the correct registration the model will be more compact. Therefore, by searching for the registration parameters which result in the most parsimonious stochastic model, we can register sequences where there is only stochastic rigidity. *The paper describes an implementation of this scheme and shows results on a number of example sequences.*

1. Introduction

A common requirement for film and broadcast is the overlaying of computer-generated imagery on live video footage. This is particularly popular in sports broadcasting: for example advertising is superimposed onto the pitch; virtual lines are rendered to assist in rule adjudications. For this process to be practical when the camera is moving, it is necessary to *register* successive frames in order that the virtual augmentations remain fixed with respect to the scene, unaffected by camera movement. The past decade has seen great successes in image registration, and real-time solutions are now commercially available [27]. However, all existing techniques depend on the assumption that the scene is largely static, and certainly that the object which one wishes to augment is not deforming.

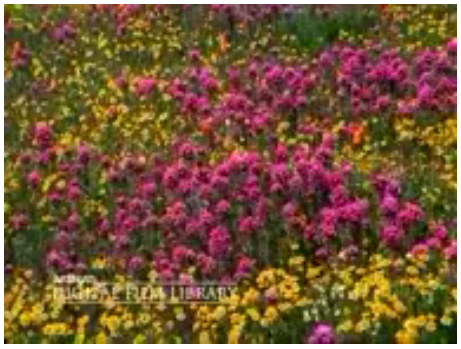
This paper considers the registration of sequences of images where the observed scene is entirely non-rigid; for example (see figure 1) a camera flying over water, a panning shot of a field of sunflowers in the wind, or footage of a crowd applauding at a sports event. In these cases, it is not possible to impose the constraint that world points have similar colour in successive views, so existing registration techniques [1, 5, 9, 11] cannot be applied. However, humans who see the video have little difficulty in decomposing the image motion into its two components: a *parametric component* introduced by the camera, and the *stochastic component* caused by water currents, wind or emotion. It is this parametric/stochastic decomposition that we seek to recover automatically from video sequences.

The first step is to define models that can represent each type of motion. For the camera motion this is straightforward: the large body of literature on the geometry of multiple views [6, 10] provides the convenient abstractions and models of projective geometry. In this paper, we generally employ a 2D perspective mapping (homography) between the image planes. For the stochastic component, we have access to the wide literature on time series analysis [16, 23]. Here, we generally consider autoregressive moving average (ARMA) processes with differencing (ARIMA).

The key to this work is the time series model, and in particular the idea of *video textures* [21, 24]. A video texture is a time series description of a video sequence which

"Crowd"

"Water"



"Flowers"

Figure 1: **Example image sequences**. "Crowd": static camera, complex motion. We wish to build a convincing videotexture. See section 2. "Water": known camera motion, planar scene. We wish to compute a single parameter; the water level. See section 3. "Flowers": panning camera, flowers blowing in a gusty wind. We wish to recover the camera pan at each frame, see section 4.

can be *driven* by random noise to generate arbitrarily long sequences of similar images, with dynamics which are consistent with those in the original clip.[1] The key idea of this paper is the following:

> Video textures are harder to learn when the camera is moving than when it is stationary

If the camera is panning, for example, the autoregressive model must learn the non-stationary parametric motion, for which it is ill-suited. On the other hand, if the images are registered to remove the camera motion, the AR model will fit well. We can measure the concept of "harder to learn" automatically by looking at the number of stochastic model parameters required to model the sequence. Therefore, in order to recover the registration, we search for the parameters that give the most efficient time series description. The constant brightness assumption is generalized to *stochastic rigidity*.

After a brief presentation of relevant work, the paper develops a new technique for videotexture construction, looking just at static scenes. Then the simplest possible example of a dynamic scene is considered, where only one parameter must be estimated. Finally, we consider the more general case of a panning camera, and provide an algorithm which can recover the camera motion at each frame. We conclude by sketching the potential for future work, including the simultaneous recovery of general scene structure and camera motion from nowhere-rigid scenes.

1.1. Image registration

An image is a 2D array of values $I(x, y)$. We are given a collection of images $I_{1..n}$. Without loss of generality, assume each image I_t is related to the first (I_1) by an (unknown) geometric transformation of pixel positions $T_t : (x, y) \rightarrow (x', y')$, so that the pixels $I_1(x, y)$ and $I_t(x', y')$ are samples from the same world point taken at different times. The task of image registration is to recover the set of transformations $T_{2..n}$.

The transformations considered in this paper are 2D projective transformations of the image. We represent 2D points in homogeneous coordinates $(x, y, 1)$ and then T_t is represented by a 3×3 homography or collineation:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

where the equality is up to scale.

There are often considered to be two approaches to registration, "direct" and "feature-based" methods. References [14] and [26] are just two examples. This paper is most similar to direct methods, and indeed, if presented with a static scene, the algorithm in §4 reduces to a (more expensive) direct correlation technique. As illustrated in figure 2, the task of the direct method is to choose transformation parameters T_t which make the set of transformed samples at each pixel $(I_t(T_t(x, y)))$ as similar as possible. This can be expressed as an error function $\epsilon(T_1, \ldots, T_n)$ which is minimized to find the optimum values of the transformation parameters. One such error function is the sample variance of each pixel over time:

$$\epsilon(T_1, \ldots, T_n) = \sum_{(x,y)} var(\{I_t(T_t(x, y))\}_{t=1}^{n})$$

where $var(i_1 \ldots i_n) = \frac{1}{n} \sum_t (i_t - \frac{1}{n} \sum i)^2$.

In stochastic rigidity we (at least conceptually) replace the var function with one which measures the deviation

[1] See movie `flowers.mpg` on site
`http://www.robots.ox.ac.uk/~awf/iccv01`

(a) Unregistered image stack. 3D points move within the image.

(b) Registered image stack Images are moved so 3D points remain fixed

Figure 2: **Image registration**. (a) The input video sequence viewed as a stack of images. The images of 3D points move within the frame. The task of registration is to (b) transform each image I_t by a transform T_t so that 3D points are aligned. For a *static scene* model, pixels in successive layers then have the same colour. Finding T_t so that the difference in colour is minimized yields the registration. In *stochastic rigidity* (SR), pixels in successive layers are related by a compact time-series model.

from a time series model. Replacing the var function here is similar in spirit to the work on space carving [4, 15] where voxel consistency is measured as deviation from a lighting model. It also has the flavour of the early mutual information work [28] where the error metric for image registration is based on deviation from causality of the scattergram of transformed intensities.

1.2. Time series analysis

Computer vision has recently seen increased use of the tools of time series analysis [2, 3, 13, 17, 18, 24], which model the temporal evolution of physical systems. A time series is a sequence of vector-valued observations $\{\mathbf{x}_t\}_{t=1}^{n}$. One task of time series analysis is to *forecast* the value of $\mathbf{x}_{t+1}$, given the previously observed values $\mathbf{x}_{1\ldots t}$. If $\mathbf{x}$ is produced by an *autoregressive process*, then the forecast is a linear combination of some number, p, of previous values:

$$\mathbf{x}_{t+1} = \mathsf{A}_0\mathbf{x}_t + \cdots + \mathsf{A}_p\mathbf{x}_{t-p} + \mathbf{w}_{t+1}$$

where the A_t are $r \times r$ ($\mathbf{x} \in \mathbb{R}^r$) matrix constants which characterize the sequence. The term $\mathbf{w}_{t+1}$ is drawn from a noise distribution, typically taken to be Gaussian. To synthesize from an AR model, choose the first p values arbitrarily, and then repeatedly predict $\mathbf{x}_{t+1}$ as above, adding noise drawn from $\mathbf{w}$'s distribution.

Conversely, if we have a sequence of observed values which we can believe to be taken from an autoregressive (AR) process of *order p*, we can estimate values of the AR parameters $\mathsf{A}_0 \ldots \mathsf{A}_p$ and the parameters of $\mathbf{w}$'s distribution, for example its covariance matrix. Algorithms which can compute these parameters from a sequence of sampled data include Yule-Walker and a family of maximum likelihood estimators [23]. In this work, Yule-Walker estimators were used, although we plan to test MLE shortly. The choice of p can be made by model selection, e.g. AIC [25].

Recently, the work of Frey *et al.* [12, 7] has considered the problem of learning from unregistered sequences, and therefore solves a problem similar to that discussed here. An important difference between their work and ours is that they model the parametric motion as a discrete set of stochastic motions, rather than parametrically as is done here. This means they are limited to simple models such as translation, and to small motions (or more accurately, to a small number of discretely sampled motions). However, their technique benefits from being amenable to an EM implementation, so it is faster, within its domain, than the algorithm in section 4.

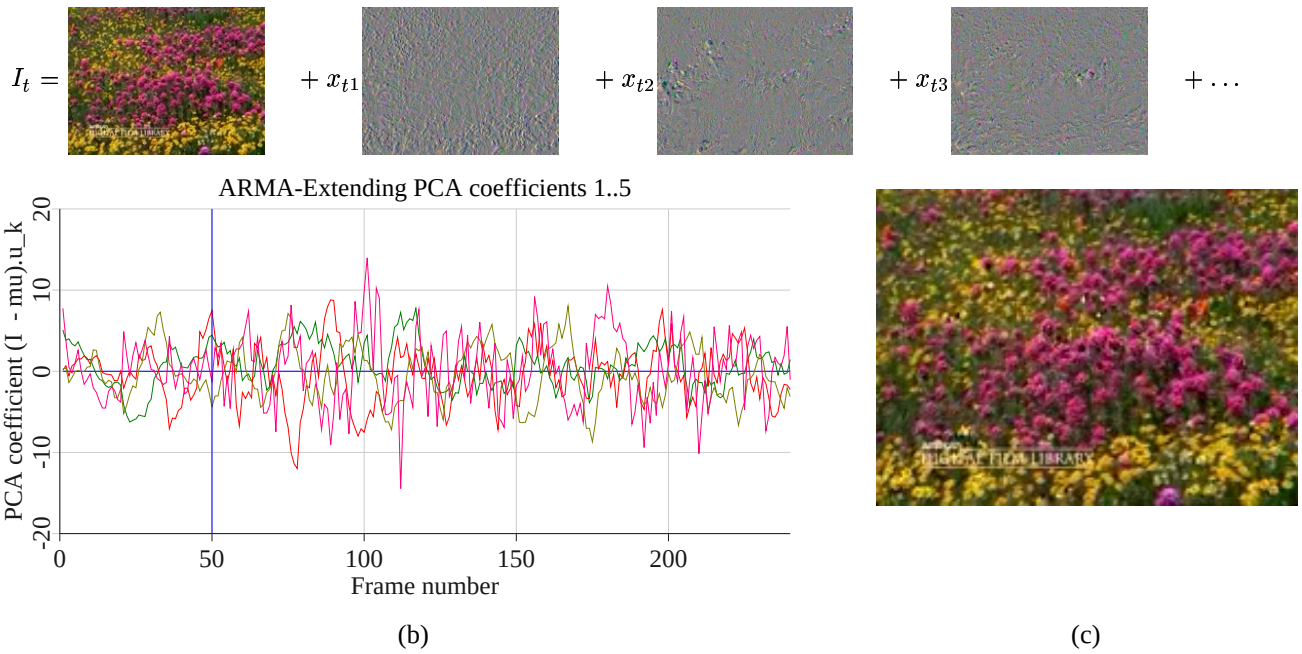
1.3. Video textures

Schodl *et al.* [21] describe the concept of a video texture: a means of generating arbitrarily long video sequences which have the behaviour of a given example. Although the goal of this paper is not just to generate video textures, the work improves a little on theirs. Their technique limits the output sequence to be made of copies of individual frames from the input sequence, and cannot therefore synthesize motions which interpolate those in the original sequence. Their model for frame transition is also discrete, meaning that dynamics need to be added post-hoc. In contrast, the current method models the temporal behaviour as samples of an underlying continuous process, with longer than 1 frame of memory. Therefore the correct dynamics emerge automatically.

Earlier, Szummer and Picard [24, 17] used a spatiotemporal autoregressive (STAR) model to describe textures such as water and steam. Their work finds an AR description of how pixels in the sequence evolve as a function of a window of pixels neighbouring in time and space, and is fitted directly to the raw image data. This model is effective for spatially coherent textures, but breaks down when there are independent motions in the image. One could imagine fitting a cluster of STAR models, but that was not explored here. The model in this paper is complementary to this—a global description of the sequence is first extracted, and then the AR model is fitted.

2. Static camera: Learning video textures

In order to address the registration problem in the later sections of the paper, we describe how we generate video textures. Several characteristics of the AR model make it an attractive technology for this purpose: it is cheap to fit, easy to run simulations from, and can model a wide range of



(b) (c)

Figure 3: **Videotextures**. (a) An orthogonal decomposition of the image sequence into principal components. The second and higher components tend to concentrate on independently moving regions. We need about 25 components to get a good reconstruction. (b) Frames 0–50 (left of the vertical blue line): the first 5 PCA coefficients $x_{1..5}$ for each frame of the input sequence. Frames 51–250: An AR(10) synthesis of PCA coefficients. (c) Frame 200 of the sequence made by recomposing the synthetic coefficients.

physical processes. In order to render the technique computationally feasible, the sequences are first subjected to data reduction by a principal components analysis (PCA), and then expressed in terms of a number of PCA coefficients. Thus each image I_t is represented by, say, 25 coefficients, stored in the vector $\mathbf{x}_t$. Figure 3 shows the procedure applied to 50 frames of the "flowers" sequence in which there is no pan.

2.1. Data reduction

In detail, we are given a set of n images I_t each of size $w \times h$, which are stored as $wh \times 1$ vectors v_t. Compute the mean image $\mu = \frac{1}{n}\sum v_t$. Assemble the $wh \times n$ matrix D with columns $d_t = v_t - \mu$. Compute the singular value decomposition $\mathsf{D} = \mathsf{USV}^\top$. Then the principal components are the columns of U and the PCA coefficients of original image t are the elements of the t^{th} row of

$$\mathsf{X} = \mathsf{VS}$$

In the type of sequences considered here, the principal components do not tend to show a sharp dropoff at any particular number. As we use PCA purely as a data reduction procedure, this is not significant, so we simply choose a fixed number of coefficients. Typically, 25 will be enough to reconstruct the sequence, and the matrix X is trimmed to keep only the first 25 columns. Examining the leading components of the PCA is interesting. As can be seen in figure 3, the analysis tends to segment independent motions in the scene, choosing one or two components for each region that is moving. Of course, a more robust decomposition than PCA, perhaps K-PCA [22] or ICA [19], would enhance this effect and potentially improve performance.

2.2. AR model fitting

Presenting the trimmed matrix of coefficients, X, to Neumaier and Schneider's `ARfit` algorithm [20], we obtain an estimate of the AR parameters $\hat{\mathsf{A}}_{1..p}$ and the noise covariance $\hat{\mathsf{C}}$. The model order may be chosen automatically using AIC, or input by the user. In the example in figure 3 the order was manually set to 10.

2.3. Removal of non-stationarity

As noted early and prominently in every textbook on time series analysis, autoregressive models can only be estimated for *stationary* time series, where the mean and variance of the noise process does not change over time. Before fitting the model, this generally means examining the sequence

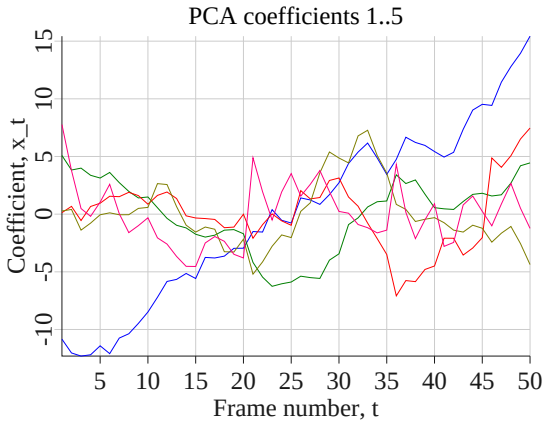


Figure 4: **Nonstationarity**. The first 5 coefficients of the "flowers" sequence, as a function of frame number t. The first component (blue) shows nonperiodic behaviour, and is removed from the AR fit in order to generate the videotexture.

and removing any nonperiodic "trends", for example by differencing or polynomial fitting. In the PCA decomposition, such non-stationarity is evident as nonperiodic coefficients of the first components. For example, in figure 4 the coefficient of the first component of the "flowers" sequence is seen to increase monotonically through the sequence. Examining the first component, it is similar to the I_x spatial derivative of the sequence. This is found to correspond to a very small amount of pan (amounting to a few pixels in the image) in the camera when taking the sequence. Simply removing the coefficients from the time series X is enough to give a good fit.

2.4. Videotexture synthesis

Finally, we can drive the AR process to generate the video texture. With $\mathbf{x}_{1..50}$ the rows of X, we generate the first synthetic frame $\mathbf{x}_{51}$ as

$$\mathbf{x}_{51} = \hat{\mathsf{A}}_0\mathbf{x}_{50} + \cdots + \hat{\mathsf{A}}_{10}\mathbf{x}_{40} + \mathbf{w}$$

where $\mathbf{w}$ is a 25×1 noise vector drawn from a multivariate Gaussian of covariance $\hat{\mathsf{C}}$. Repeating this process generated the sequence of coefficients in figure 3b, and the example movie `flowers.mpg`. In this movie, the first 50 frames are reconstructions of frames from the original sequence, and the next 200 are synthetic. The synthetic frames are entirely believable, and include the same independent motions of different parts of the scene, but in different combinations.

```
function error = epsilon_SR(H_1, ..., H_n)
   for each image I_t
      Compute registered image I'_t = warp(H_t, I_t)
   end

   Compute PCA coefficients X from warped images I'_t.
   Discard nonstationary columns of X.
   Estimate AR parameters, and noise covariance Ĉ.
return − log det(Ĉ) + λr²p
```

Figure 6: **Stochastic Rigidity**. Given a set of putative registering homographies, the AR model is estimated which best models the warped images. The SR metric is the information content of the AR model.

3. Estimation from non-rigid scenes

The previous section dealt with a static scene, and showed how to learn an autoregressive model of the image generation process. In this section we come to the heart of the paper, the use of this model to solve registration problems. To that end we have chosen a simple, but real, problem in which only one parameter is to be estimated. The "water" sequence in figure 1 is a test shot for a movie special effect. The cameraman is walking along a bridge, looking into the water, and the goal is to generate a composited sequence in which a computer-generated object appears to float on the water. The 3D path of the camera relative to the bridge can be computed by tracking points on the bridge, so the 3D camera positions are available for each frame. Thus, all that is required to complete the task is an estimate of h, the distance of the water below the bridge.

Assuming the water surface to be planar, its motion under arbitrary camera movement will be described by a 2D projective transformation, or homography. Registering the images of the water surface entails recovering the homographies that relate each successive pair of images. Given the camera path, these homographies are all specified by the single unknown parameter h. In order to find h we require an error metric $\epsilon(h)$ which is minimized when the images are correctly registered.

For a time-varying scene, such an error metric can be computed by fitting an AR model to the registered images and evaluating the efficiency of the fit. Based on the principle that video textures are harder to learn from unregistered training sequences, the correctly registered sequence will have a more efficient AR representation. The value of h that yields the most compact AR model is then what is sought.

Computation of information content of an AR model is a common requirement in statistics, and several measures have been proposed [20, 23]. All amount to summing terms which count the number of parameters in the model and the entropy of the noise process. Thus, if the number of princi-

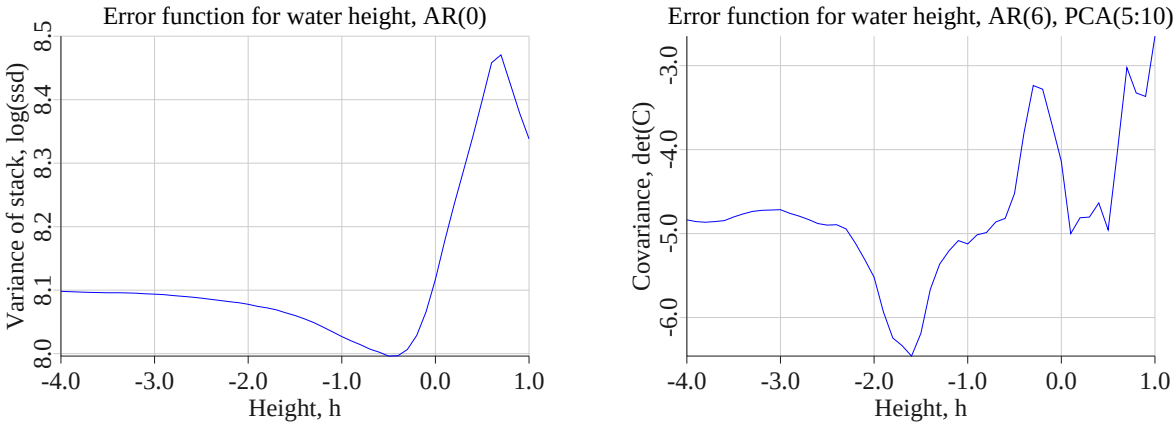


Figure 5: **Estimation of water height**. The ground truth water level is $Z = -1.5 \pm 0.1$ units—the bridge platform is at $Z = 0$. The error function is sampled at 0.1 unit intervals, and the value of the error plotted. (SSD) The SSD minimum occurs far from the ground truth. (SR) The SR minimum occurs at the ground truth position. At this point, the registered water surface images are well modelled by an AR(6) process applied to 5 principal components. The graph is largely insensitive to the choice of these parameters of model order and number of components.

pal components retained is r, the SR criterion is given by

$$\epsilon_{SR} = \lambda p r^2 - \log\det \hat{\mathsf{C}}$$

where λ is the number of bits required to represent an element of the matrices A. In order to avoid the delicate issue of specifying λ, the models in this work have a preset order. Therefore the number of parameters does not vary during the search for the optimal registration, and so the only term of interest is the entropy of the noise process. Thus, we can compare two registrations by looking at the estimated covariance matrix $\hat{\mathsf{C}}$ of the AR noise process. The computation is summarised in figure 6.

Armed with this metric, we may now state the algorithm for estimation of water height in our example shot. The 3D coordinate system is arranged so that the $Z = 0$ plane corresponds to the platform. Thus, assuming the water surface is parallel to the platform, the water is in the plane $Z = h$, where h is the unknown height.

For concreteness, the sequence will be registered to image 0, in which a rectangular region of interest of size about 128^2 pixels is defined. The rectangle contains only water, so the image sequence is indeed "nowhere-static". In particular it contains a standing wave caused by flow past a bridge pylon, which gives rise to periodic intensity variation.

Because we have only one parameter to find, an exhaustive search for h is possible. For each value in a range, the set of homographies mapping each image to frame 0 is computed from the known camera positions, and the SR criterion is evaluated. The results are graphed in figure 5, in which it is seen that the SR minimum does indeed fall at the ground-truth value. For comparison, the traditional SSD metric is also shown, which has a unique, but erroneous minimum. The SSD is easily computed within this scheme, as it results from "fitting" a zero-order ($p = 0$) AR model to the sequence. Thus the SR criterion subsumes the static scene assumption.

3.1. Computational cost

The computational cost of SR estimation is greater than SSD, because of the need to compute a PCA for each parameter estimate. The two major components of the cost are the image warping (which is common to SSD) and the principal components analysis. The image warping time is linear in the size of the images and the length of the sequence, for this example it averaged 15 seconds. The PCA, on the other hand, takes time which is linear in the image size, but quadratic in the length of the sequence. In this example, the PCA executed in 6 seconds on average, but this would increase rapidly for longer sequences. A number of strategies present for amelioration of this situation, such as windowing the PCA, or performing data reduction to a wavelet or Fourier basis. Currently these remain areas for future work.

4. Registration under camera rotation

Moving from a single-parameter estimation problem to the more general case of a moving camera is relatively straightforward. The SR error function of the previous section is defined in terms of the set of homographies, and so is equally

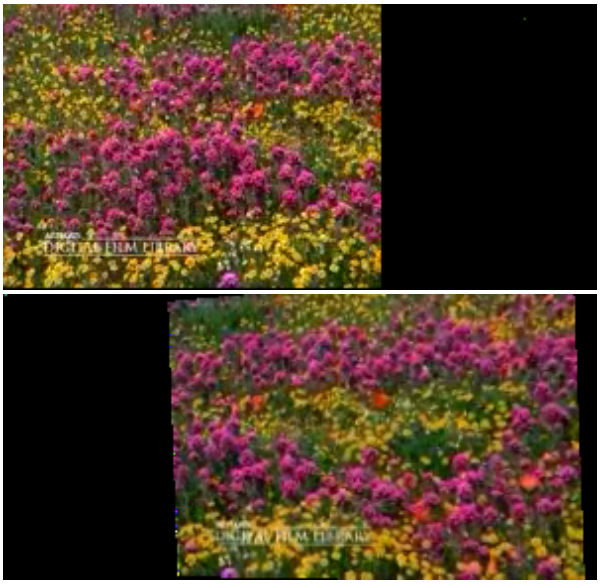


Figure 7: **Registration of pan sequence**. The first and last frames of the pan sequence, registered into a common frame.

applicable in the general case. In general, then, a nonlinear minimization of ϵ_{SR} over all homographies, with a good initial estimate, will find the optimal registration. Of course, the hope is that the initial estimate does not have to be so close that it is impossible to obtain. In this, figure 5 is encouraging: for that problem, the basin of attraction of the true minimum runs from 50% to 200% of the true value.

As an example of a problem with more parameters, we consider again the "flowers" scene. We have a 101 frame panning shot from which it is desired to compute the pan. Treated as a general registration problem, there are 800 parameters to estimate, 8 for each inter-frame homography. Knowing, however, that the sequence is a single-axis pan, the relationship between any view and the reference view may be written

$$H_t = \mathsf{K} e^{\theta_t [\mathbf{r}]_\times} \mathsf{K}^{-1}$$

where the matrix K holds the intrinsic camera parameters [10], the unit vector $\mathbf{r}$ is the pan axis—unknown, but constant through the sequence—and θ_t represents the pan angle at frame t. Thus, if there are k camera intrinsics, and n views, then the total number of parameters to be estimated is $k + 2(\text{for } \mathbf{r}) + (n-1)$.

In figure 7, only the focal length is unknown, so the number of parameters is 103. An initial estimate was obtained by manually selecting four points in the first and last images and fitting a homography between the two. Applying self calibration [10] yields an estimate of the calibration K, and the rotation axis $\mathbf{r}$ as well as the total pan angle. Linearly interpolating the angles gives an initial homograpy estimate for each frame. Although the registered sequence (see movie `panxx.mpg`) shows significant misregistration, this is a good initial estimate. Then it is a simple matter to begin a nonlinear minimization. In this example, a modified Levenberg Marquardt algorithm [8] was used, with finite difference derivatives. A typical 6 iterations requires about 600 function evaluations. The minimizing parameters were used to create a stabilized sequence (in `panxx2.mpg`), which shows greatly improved registration over the initial estimate.

5. Discussion

The paper has addressed the problem of computing geometric quantities such as camera motion from images of natural scenes—where no part of the scene is rigid, but there is nevertheless a clear decomposition into parametric and stochastic motions. The primary area of application, in post production, means that algorithms are needed which can achieve human levels of competence at these tasks. The paper contributes the novel concept of "stochastic rigidity" to allow such problems to begin to be addressed.

This model is sufficiently powerful to allow the estimation of an 100-parameter image registration problem, while being of sufficiently wide scope to model processes such as water, and flowers blowing in the wind. It can be seen as a generalization of the brightness constancy assumption, and similar generalizations have recently emerged in other contexts.

Of course, as with rigid-scene reconstruction, there is the potential for ambiguity in SR-guided reconstructions. For example, in section 3 the water is moving from right to left across the scene. An unconstrained homography registration would follow the water surface, which might not be desirable. In the case addressed here, the constraints provided by the known camera motion eliminate the ambiguity. In other cases, external constraints will need to be imposed to regularize the problem.

The data reduction before AR fitting is the weakest part of the system. On the crowd scene, it fails to usefully decompose the motion, so the generated videotextures acquire undesirable ringing artefacts. We are confident that these will be eliminated by using a more sophisticated decomposition. Although the specific data reduction strategy chosen is limited, it appears that the principle of first acquiring a set of global descriptors is a significant benefit of our method.

Other areas for further work include: automatic identification and removal of nonstationarity in the principal components; and comparison of STAR and PCA-AR fitting in the SR criterion.

In conclusion, the paper has shown that stochastic modelling of video sequences can be used for estimation prob-

lems, and significantly extends the scope of scenes that can be analyzed in computer vision.

References

[1] M. J. Black and P. Anandan. A framework for the robust estimation of optical flow. In *Proc. ICCV*, pages 231–236, 1993.

[2] M. Brand. Shadow puppetry. In *Proc. ACM SIGGRAPH*, 2000.

[3] C. Bregler. Learning and recognising human dynamics in video sequences. In *Proc. CVPR*, Jun 1997.

[4] A. Broadhurst and R. Cipolla. A statistical consistency check for the space carving algorithm. In *Proc. BMVC*, pages 282–291, 2000.

[5] D. Capel and A. Zisserman. Automated mosaicing with super-resolution zoom. In *Proc. CVPR*, pages 885–891, Jun 1998.

[6] O. D. Faugeras. *Three-Dimensional Computer Vision: a Geometric Viewpoint*. MIT Press, 1993.

[7] B. Frey and N. Jojic. Learning mixture models of images and inferring spatial transformations using the EM algorithm. In *Proc. CVPR*, Jun 1999.

[8] P. E. Gill and W. Murray. Algorithms for the solution of the nonlinear least-squares problem. *SIAM J Num Anal*, 15(5):977–992, 1978.

[9] F. Glazer, G. Reynolds, and P. Anandan. Scene matching by hierarchical correlation. In *Proc. CVPR*, pages 432–441, 1983.

[10] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049, 2000.

[11] M. Irani, B. Rousso, and S. Peleg. Computing occluding and transparent motions. *IJCV*, 12(1):5–16, 1994.

[12] N. Jojic, N. Petrovic, B. J. Frey, and T. S. Huang. Transformed hidden Markov models: Estimating mixture models of images and inferring spatial transformations in video sequences. In *Proc. CVPR*, 2000.

[13] A. Kokaram and S. Godsill. A system for reconstruction of missing data in image sequences using sampled 3D AR models and MRF motion priors. In *Proc. ECCV*, pages 613–624, 1996.

[14] R. Kumar, P. Anandan, M. Irani, J. Bergen, and K. Hanna. Representation of scenes from collections of images. In *ICCV Workshop on the Representation of Visual Scenes*, 1995.

[15] K. Kutulakos and S. Seitz. A theory of shape by space carving. In *Proc. ICCV*, pages 307–314, 1999.

[16] A. Papoulis. *Probability, Random Variables and Stochastic Processes*. Mc Graw Hill, New York, 1991.

[17] R. Picard. A society of models for video and image libraries. Technical Report 360, MIT Media Lab, 1996.

[18] J. Rittscher and A. Blake. Classification of human body motion. In *Proc. ICCV*, pages 634–639, 1999.

[19] S. Roberts and R. Everson. *Independent Components Analysis: Principles and Practice*. Cambridge University Press, 2001.

[20] T. Schneider and A. Neumaier. Algorithm: ARfit — a Matlab package for estimation and spectral decomposition of multivariate autoregressive processes. *ACM TOMS*, page (to appear), 2000.

[21] A. Schodl, R. Szeliski, D. H. Salesin, and I. Essa. Video textures. In *Proceedings of SIGGRAPH*, pages 489–498, 2000.

[22] B. Sch olkopf, A. Smola, and K.-R. M uller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

[23] R. H. Shumway and D. S. Stoffer. *Time Series Analysis and its Applications*. Springer, 2000.

[24] M. Szummer. Temporal texture modeling. Master's thesis, MIT Media Lab, Cambridge MA, May 1995.

[25] P. H. S. Torr. An assessment of information criteria for motion model selection. In *Proc. CVPR*, Jun 1997. To appear in CVIU.

[26] P. H. S. Torr, A. W. Fitzgibbon, and A. Zisserman. The problem of degeneracy in structure and motion recovery from uncalibrated image sequences. *IJCV*, 32(1):27–44, Aug 1999.

[27] Ultimatte Corp. http://www.ultimatte.com, 2000.

[28] P. Viola and W. Wells. Alignment by maximization of mutual information. In *Proc. ICCV*, pages 16–23, Jun 1995.