Efficient Computation of the KL Divergence between Dynamic Textures

Antoni B. Chan and Nuno Vasconcelos

Statistical Visual Computing Laboratory

SVCL UCSD

SVCL-TR 2004/02 rev4

November 2004

Efficient Computation of the KL Divergence between Dynamic Textures

Antoni B. Chan and Nuno Vasconcelos
Statistical Visual Computing Lab
Department of Electrical and Computer Engineering
University of California, San Diego

November 2004

Abstract

We derive the KL divergence between dynamic textures in state space. We also derive a set of recursive equations for the calculation of the Kullback-Leibler divergence between dynamic textures in image space. The recursive equations are computationally efficient and require less memory storage than the non-recursive counterpart.

Author email: abchan@ucsd.edu

SVCL Technical reports are available on the SVCL's web page at
http://www.svcl.ucsd.edu

University of California, San Diego
Statistical Visual Computing Laboratory
9500 Gilman Drive, Mail code 0407
EBU 1, Room 5512
La Jolla, CA 92093-0407

1 Introduction

A dynamic texture is a linear dynamical system used to model a video sequence. Since the dynamic texture is a generative probabilistic model, the KL divergence can be used to compute distances between different dynamic textures. In this note, we derive the KL divergence between dynamic textures. In Section 2, we start by reviewing the probability distributions of the dynamic texture model. In Section 3, we derive the KL divergence between the state spaces of two dynamic textures. In Section 4, we define the KL divergence between the image spaces of two dynamic textures, and in Section 5, we derive a set of recursive equations for efficiently computing the image space KL divergence. The recursive equations are computationally efficient and require less memory storage than the non-recursive counterpart.

2 Dynamic Texture Model

A dynamic texture [1] is an auto-regressive process modeled by

$$x_{t+1} = Ax_t + Bv_t \tag{1}$$

$$y_t = Cx_t + w_t \tag{2}$$

where, $x_t \in \mathbb{R}^n$ is an n dimensional state vector, $y_t \in \mathbb{R}^m$ is the m dimensional image vector, $A \in \mathbb{R}^{n\times n}$ is the state transition matrix, $v_t \sim_{iid} \mathcal{N}(0, I_{n_v})$ is the n_v dimensional driving process (typically, $n \ll m$ and $n_v \leq n$) with transformation $B \in \mathbb{R}^{n\times n_v}$, $C \in \mathbb{R}^{m\times n}$ is a matrix containing the principal component vectors, $w_t \sim_{iid} \mathcal{N}(0, R)$ with $R \in \mathbb{R}^{m\times m}$ is the image noise process, and x_0 is the known initial condition. Note that $Bv_t \sim \mathcal{N}(0, Q)$ where $Q = BB^T$. We will also assume that the covariance of the image noise, R, is diagonal. A dynamic texture model is completely specified using the parameters $\Theta = \{A, B, C, R, x_0\}$.

2.1 Probability Density Functions

We first obtain the probability density functions associated with the dynamic texture. In the following we will assume that x_0 is constant. The state is governed by a Gauss Markov process [2], hence the conditional probability of the state is

$$p(x_t|x_{t-1}) = G(x_t, Ax_{t-1}, Q) \tag{3}$$

$$= \frac{1}{\sqrt{(2\pi)^n |Q|}} e^{-\frac{1}{2}\|x_t - Ax_{t-1}\|_Q^2} \tag{4}$$

where $\|x\|_Q^2 = x^T Q^{-1} x$. Recursively substituting into the state equations, we have

$$x_t = A^t x_0 + \sum_{i=1}^{t} A^{t-i} B v_i \tag{5}$$

A single state is the linear combination of Gaussian random variables, thus the probability of a single state is also Gaussian

$$p(x_t) = \mathcal{N}(\mu_t, S_t) \tag{6}$$

where

$$\mu_t = A^t x_0 \tag{7}$$

$$S_t = AS_{t-1}A^T + Q = \sum_{i=0}^{t-1} A^i Q (A^i)^T \tag{8}$$

2.1.1 State Sequence Probability

Since the driving process is Gaussian, the joint probability of a state sequence is also Gaussian. Specifically, we have

$$x_{t+k} = A^k x_t + \sum_{i=1}^{k} A^{k-i} B w_{t+i-1} \tag{9}$$

$$\mathrm{cov}(x_{t+k}, x_t) = A^k S_t \tag{10}$$

Let $x_1^\tau = (x_1, x_2, ..., x_\tau)$ be the sequence of τ state vectors, then the probability of x_1^τ is

$$p(x_1^\tau) = \mathcal{N}(\mu, \Sigma) \tag{11}$$

where

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_\tau \end{bmatrix}, \ \Sigma = \begin{bmatrix} S_1 & (AS_1)^T & (A^2 S_1)^T & \cdots & (A^{\tau-1} S_1)^T \\ AS_1 & S_2 & (AS_2)^T & \cdots & (A^{\tau-2} S_2)^T \\ A^2 S_1 & AS_2 & S_3 & \cdots & (A^{\tau-3} S_3)^T \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A^{\tau-1} S_1 & A^{\tau-2} S_2 & A^{\tau-3} S_3 & \cdots & S_\tau \end{bmatrix} \tag{12}$$

Alternatively, using conditional probability we have

$$p(x_1^\tau) = p(x_1) \prod_{i=2}^{\tau} p(x_i | x_{i-1}) \tag{13}$$

$$= \prod_{i=1}^{\tau} G(x_i, Ax_{i-1}, Q) \tag{14}$$

$$= \frac{1}{\sqrt{(2\pi)^{n\tau} |Q|^\tau}} e^{-\frac{1}{2} \sum_{i=1}^{\tau} \|x_i - Ax_{i-1}\|_Q^2} \tag{15}$$

The inverse of the covariance matrix Σ can be determined in closed-form by examining the exponent term in (15). We will first look at an example where $\tau = 3$.

$$\begin{aligned}
\sum_{i=1}^{3} \|x_i - Ax_{i-1}\|_Q^2 \quad &= \quad \begin{bmatrix} x_2 \\ x_3 \end{bmatrix}^T \begin{bmatrix} A^T Q^{-1} A & -A^T Q^{-1} \\ -Q^{-1} A & Q^{-1} \end{bmatrix} \begin{bmatrix} x_2 \\ x_3 \end{bmatrix} \qquad (16) \\
&+ \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} A^T Q^{-1} A & -A^T Q^{-1} \\ -Q^{-1} A & Q^{-1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\
&+ (x_1^T Q^{-1} x_1 - 2x_1^T Q^{-1} A x_0 + x_0^T A^T Q^{-1} A x_0)
\end{aligned}$$

$$\begin{aligned}
&= \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}^T \begin{bmatrix} A^T Q^{-1} A + Q^{-1} & -A^T Q^{-1} & 0 \\ -Q^{-1} A & A^T Q^{-1} A + Q^{-1} & -A^T Q^{-1} \\ 0 & -Q^{-1} A & Q^{-1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \\
&\quad - 2x_1^T Q^{-1} A x_0 + x_0^T A^T Q^{-1} A x_0 \\
&= (x^\tau)^T \Sigma^{-1} x^\tau - 2b^T x^\tau + c \qquad (17)
\end{aligned}$$

where

$$\Sigma^{-1} = \begin{bmatrix} A^T Q^{-1} A + Q^{-1} & -A^T Q^{-1} & 0 \\ -Q^{-1} A & A^T Q^{-1} A + Q^{-1} & -A^T Q^{-1} \\ 0 & -Q^{-1} A & Q^{-1} \end{bmatrix} \qquad (18)$$

$$b^T = \begin{bmatrix} x_0^T A^T Q^{-1} & 0 & 0 \end{bmatrix} \qquad (19)$$

$$c = x_0^T A^T Q^{-1} A x_0 \qquad (20)$$

By simply multiplying Σ^{-1} with Σ from (12) (or generally, by recursively taking the block matrix inverse of Σ^{-1}) it is easy to verify that Σ^{-1} is indeed the inverse of Σ. Completing the square on (17) we have

$$\sum_{i=1}^{3} \|x_i - Ax_{i-1}\|_Q^2 = \|x^\tau - \mu\|_\Sigma^2 + c - \mu^T \Sigma^{-1} \mu \qquad (21)$$

where $\mu = \Sigma b = \begin{bmatrix} Ax_0 \\ A^2 x_0 \\ A^3 x_0 \end{bmatrix}$ and $c - \mu^T \hat{\Sigma}^{-1} \mu = 0$, yielding

$$\sum_{i=1}^{3} \|x_i - Ax_{i-1}\|_Q^2 \quad = \quad (x^\tau - \mu)^T \hat{\Sigma}^{-1} (x^\tau - \mu) \qquad (22)$$

Hence, we have reduced the product of Gaussians in (15) into a single Gaussian and have thus found a closed-form solution to the inverse of the covariance matrix. Note that this also implies that $|\Sigma| = |Q|^\tau$. In general, the inverse of the covariance matrix is the block Toeplitz matrix,

$$\Sigma^{-1} = \begin{bmatrix} s_1 & s_2^T & 0 & \cdots & 0 \\ s_2 & s_1 & s_2^T & \cdots & 0 \\ 0 & s_2 & \ddots & \ddots & 0 \\ \vdots & \vdots & \ddots & s_1 & s_2^T \\ 0 & 0 & 0 & s_2 & Q^{-1} \end{bmatrix} \tag{23}$$

where

$$s_1 = Q^{-1} + A^T Q^{-1} A \tag{24}$$

$$s_2 = -Q^{-1} A \tag{25}$$

Thus, the inverse of the covariance matrix of the state sequence probability is easily computed, thereby avoiding the inversion of a potentially very large matrix.

2.1.2 Image Sequence Probability

Let $y_1^\tau = (y_1, y_2, ..., y_\tau)$ be the sequence of τ image vectors, then the probability of y_1^τ is

$$p(y_1^\tau) = \mathcal{N}(\gamma, \Phi) \tag{26}$$

where

$$\gamma = \mathbf{C}\mu, \; \Phi = \mathbf{C}\Sigma\mathbf{C}^T + \mathbf{R} \tag{27}$$

$$\mathbf{C} = \begin{bmatrix} C & 0 & \cdots & 0 \\ 0 & C & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & C \end{bmatrix}, \; \mathbf{R} = \begin{bmatrix} R & 0 & \cdots & 0 \\ 0 & R & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & R \end{bmatrix} \tag{28}$$

Unfortunately, y_1^τ is $m\tau$-dimensional, where m is the number of pixels in the image and τ is the length of the image sequence, thus direct evaluation is computationally intractable. For example, suppose we have an image patch of 48×48 pixels over 20 frames, then the sequence vector y_1^{20} will be 46,080 dimensional. The covariance matrix of the Gaussian will have over 2 billion elements, requiring 15.8 GB using double precision floating point.

3 KL Divergence between Dynamic Textures in State Space

The KL divergence rate between two random processes with distributions, $p(X)$ and $q(X)$ over $X = (x_1, x_2, \ldots)$, is defined as

$$D(p(X)\,\|q(X)) = \lim_{t\to\infty} \frac{1}{\tau} D(p(x_1^\tau)\,\|q(x_1^\tau))\,. \tag{29}$$

Given that $p(x)$ and $q(x)$ are distributions of Markov processes, (29) can be simplified using the chain rule of divergence [4],

$$D(p(x_1^\tau)\,\|q(x_1^\tau)\,) = D(p(x_1)\,\|q(x_1)\,) + \sum_{i=2}^{\tau} D(p(x_i|x_{i-1})\,\|q(x_i|x_{i-1})\,) \tag{30}$$

Let $p(x_1^\tau)$ and $q(x_1^\tau)$ be the probability distributions of the state sequence $x_1^\tau = (x_1, \cdots, x_\tau)$ of two dynamic textures parameterized by (A_1, Q_1, x_{01}) and (A_2, Q_2, x_{02}). The KL divergence of the initial state vector is

$$D(p(x_1)\,\|q(x_1)\,) = \frac{1}{2}\,\|A_1 x_{01} - A_2 x_{02}\|_{Q_2}^2 + \frac{1}{2}\log\frac{|Q_2|}{|Q_1|} + \frac{1}{2}\mathrm{tr}(Q_2^{-1}Q_1) - \frac{n}{2} \tag{31}$$

and the conditional KL term is

$$\begin{aligned}
&D(p(x_i|x_{i-1})\,\|q(x_i|x_{i-1})\,)\\
&= \int p(x_{i-1}) \int p(x_i|x_{i-1}) \log\frac{p(x_i|x_{i-1})}{q(x_i|x_{i-1})} dx_i dx_{i-1}\\
&= \int p(x_{i-1}) \int G(x_i, A_1 x_{i-1}, Q_1) \log\frac{G(x_i, A_1 x_{i-1}, Q_1)}{G(x_i, A_2 x_{i-1}, Q_2)} dx_i dx_{i-1}\\
&= \int p(x_{i-1}) \frac{1}{2}\left[\|(A_1 - A_2)x_{i-1}\|_{Q_2}^2 + \log\frac{|Q_2|}{|Q_1|} + \mathrm{tr}(Q_2^{-1}Q_1) - n\right] dx_{i-1}\\
&= \frac{1}{2}\left[\mathrm{tr}(\bar{A}^T Q_2^{-1}\bar{A}(S_{i-1} + \mu_{i-1}\mu_{i-1}^T)) + \log\frac{|Q_2|}{|Q_1|} + \mathrm{tr}(Q_2^{-1}Q_1) - n\right]
\end{aligned}$$

where $\bar{A} = A_1 - A_2$, and in the last line we have used the property that if $p(x)$ has mean μ and covariance Σ,

$$\begin{aligned}
\int p(x)\,\|Ax\|_B^2\,dx &= \mathrm{E}[x^T A^T B^{-1} A x]\\
&= \mathrm{E}[\mathrm{tr}(A^T B^{-1} A x x^T]\\
&= \mathrm{tr}(A^T B^{-1} A \mathrm{E}[x x^T])\\
&= \mathrm{tr}(A^T B^{-1} A(\Sigma + \mu\mu^T))
\end{aligned}$$

Finally, summing over the conditional KL terms, the KL divergence on the RHS of (29) is

$$\frac{1}{\tau} D(p(x_1^\tau)\,\|q(x_1^\tau)\,) = \tag{32}$$

$$\frac{1}{2}\left[\log\frac{|Q_2|}{|Q_1|} + \mathrm{tr}(Q_2^{-1}Q_1) - n + \frac{1}{\tau}\left\|A_1x_{01} - A_2x_{02}\right\|^2_{Q_2}\right.$$

$$\left. + \frac{1}{\tau}\sum_{i=2}^{\tau}\mathrm{tr}\left(\bar{A}^TQ_2^{-1}\bar{A}(S_{i-1} + \mu_{i-1}\mu_{i-1}^T)\right)\right]$$

where $\bar{A} = A_1 - A_2$, and S_{i-1} and μ_{i-1} are the covariance and mean associated with the state x_{i-1} of the first dynamic texture.

4 KL Divergence between Dynamic Textures in Image Space

Let $p_1(y_1^\tau)$ and $p_2(y_1^\tau)$ be the probability density functions of an image sequence for two texture models parameterized by Θ_1 and Θ_2, respectively. The KL divergence rate [3] between the two textures models is defined as,

$$D(p_1\,\|p_2\,) = \lim_{\tau\to\infty}\frac{1}{\tau}D(p_1(y_1^\tau)\,\|p_2(y_1^\tau)\,) \tag{33}$$

Since p_1 and p_2 are both Gaussian, there is a closed-form solution of the KL divergence for length τ given by

$$D(p_1\,\|p_2\,) = \frac{1}{2}\left[\log\frac{|\Phi_2|}{|\Phi_1|} + \mathrm{tr}\left(\Phi_2^{-1}\Phi_1\right) + \left\|\gamma_1 - \gamma_2\right\|^2_{\Phi_2} - m\tau\right] \tag{34}$$

Direct evaluation of the KL is computationally intractable, since the formula depends on Φ_1 and Φ_2, which are both very large covariance matrices.

5 Recursive Evaluation of KL Divergence

While direct computation of the image covariance matrix Φ is intractable, it is possible to rewrite the terms of the KL divergence into a recursive form by using several matrix identities. The resulting formulation reduces the required memory and is computationally efficient.

We will now derive the recursive equations for each of the terms in KL divergence equation (34) for time τ given time $\tau-1$. We will refer to matrices (and vectors) at time τ as A_i^τ, and at time $\tau-1$ as $A_i^{\tau-1}$, where i is the index for p_1 or p_2. For simplicity, we will also refer to the image at the current time step τ as y, and the sequence of preceding $\tau-1$ images as Y. The covariance and means of p_1 and p_2 can be defined recursively as

$$\gamma_1^\tau = \begin{bmatrix}\gamma_{1Y}\\ \gamma_{1y}\end{bmatrix},\;\; \Phi_1^\tau = \begin{bmatrix}\Phi_{1YY} & \phi_{1Yy}\\ \phi_{1yY} & \phi_{1yy}\end{bmatrix} \tag{35}$$

$$\gamma_2^\tau = \begin{bmatrix}\gamma_{2Y}\\ \gamma_{2y}\end{bmatrix},\;\; \Phi_2^\tau = \begin{bmatrix}\phi_{2YY} & \phi_{2Yy}\\ \phi_{2yY} & \phi_{2yy}\end{bmatrix} \tag{36}$$

Similarly, we can define $\mu_{1x}, \mu_{1X}, \Sigma_{1XX}, \Sigma_{1Xx}, \Sigma_{1xX}, \Sigma_{1xx}$ for the probability of a state sequence under p_1, and likewise for p_2.

5.1 Mahalanobis Distance Term

For the Mahalanobis distance, we have the following recursion,

$$\|\gamma_1^\tau - \gamma_2^\tau\|^2_{\Phi_2^\tau} = \left\|\gamma_1^{\tau-1} - \gamma_2^{\tau-1}\right\|^2_{\Phi_2^{\tau-1}} + \|z\|^2_{\hat{\Phi}_2} \tag{37}$$

where $\|z\|^2_{\hat{\Phi}_2}$ is the update term with

$$z = \phi_{2yY}(\Phi_2^{\tau-1})^{-1}(\gamma_1^{\tau-1} - \gamma_2^{\tau-1}) - (\gamma_{1y} - \gamma_{2y}) \tag{38}$$

$$\hat{\Phi}_2 = \phi_{2yy} - \phi_{2yY}(\Phi_2^{\tau-1})^{-1}\phi_{2Yy} \tag{39}$$

Substituting for the image covariance (in terms of the state covariance) and using the matrix inversion lemma,

$$\phi_{2yY}(\Phi_2^{\tau-1})^{-1} = C_2\Sigma_{2xX}\mathbf{C}_2^T(\mathbf{R}_2^{-1} - \mathbf{R}_2^{-1}\mathbf{C}_2(\beta_2^{\tau-1})^{-1}\mathbf{C}_2^T\mathbf{R}_2^{-1}) \tag{40}$$

$$= C_2\Sigma_{2xX}\Delta_2\mathbf{C}_2^T\mathbf{R}_2^{-1} \tag{41}$$

with

$$\beta_2^{\tau-1} = (\Sigma_2^{\tau-1})^{-1} + \mathbf{C}_2^T\mathbf{R}_2^{-1}\mathbf{C}_2 \tag{42}$$

$$\Delta_2 = I - (\mathbf{C}_2^T\mathbf{R}_2^{-1}\mathbf{C}_2)(\beta_2^{\tau-1})^{-1} \tag{43}$$

The update covariance matrix becomes

$$\hat{\Phi}_2 = (C_2\Sigma_{2xx}C_2^T + R_2) - (C_2\Sigma_{2xX}\Delta_2\mathbf{C}_2^T\mathbf{R}_2^{-1})\mathbf{C}_2\Sigma_{2Xx}C_2^T \tag{44}$$

$$= C_2\Gamma_2C_2^T + R_2 \tag{45}$$

where

$$\Gamma_2 = \Sigma_{2xx} - \Sigma_{2xX}\Delta_2(\mathbf{C}_2^T\mathbf{R}_2^{-1}\mathbf{C}_2)\Sigma_{2Xx} \tag{46}$$

and the inverse of $\hat{\Phi}_2$ can be taken by using the matrix inversion lemma,

$$\hat{\Phi}_2^{-1} = R_2^{-1} - R_2^{-1}C_2\hat{\Gamma}_2^{-1}C_2^TR_2^{-1} \tag{47}$$

$$\hat{\Gamma}_2 = \Gamma_2^{-1} + C_2^TR_2^{-1}C_2 \tag{48}$$

Finally, the update to the Mahalanobis distance is computed as

$$z = C_2\Sigma_{2xX}\Delta_2(\mathbf{C}_2^T\mathbf{R}_2^{-1}\mathbf{C}_1\mu_1^{\tau-1} - \mathbf{C}_2^T\mathbf{R}_2^{-1}\mathbf{C}_2\mu_2^{\tau-1}) \tag{49}$$

$$- C_1\mu_{1x} + C_2\mu_{2x}$$

$$\|z\|^2_{\hat{\Phi}_2} = z^TR_2^{-1}z - (z^TR_2^{-1}C_2)\hat{\Gamma}_2^{-1}(C_2^TR_2^{-1}z) \tag{50}$$

The computation of the distance requires the inverse of Γ_2 and $\hat{\Gamma}_2$, both $n \times n$ matrices, and β_2, an $n(\tau-1) \times n(\tau-1)$ matrix. Fortunately, the inverse of the β_2 matrix can be computed efficiently using recursion.

5.2 Inverse of Beta Matrix

We will now derive a recursive expression for inverting β. Let $\beta^\tau = (\Sigma^\tau)^{-1} + \mathbf{C}^T\mathbf{R}^{-1}\mathbf{C}$. The inverse of Σ^τ can be expressed recursively,

$$(\Sigma^\tau)^{-1} = \begin{bmatrix} s_1 & s_2^T \quad 0 \\ \begin{matrix} s_2 \\ 0 \end{matrix} & (\Sigma^{\tau-1})^{-1} \end{bmatrix} \tag{51}$$

and thus,

$$\beta^\tau = \begin{bmatrix} s_1 + C^T R^{-1} C & s_2^T \quad 0 \\ \begin{matrix} s_2 \\ 0 \end{matrix} & \beta^{\tau-1} \end{bmatrix} \tag{52}$$

where s_1 and s_2 are defined in (24) and (25). Taking the inverse of the block matrix, we have

$$(\beta^\tau)^{-1} = \begin{bmatrix} 0 & 0 \\ 0 & (\beta^{\tau-1})^{-1} \end{bmatrix} + \begin{bmatrix} I \\ U_\tau \end{bmatrix} V_\tau^{-1} \begin{bmatrix} I & U_\tau^T \end{bmatrix} \tag{53}$$

$$= \begin{bmatrix} V_\tau^{-1} & V_\tau^{-1} U_\tau^T \\ U_\tau V_\tau^{-1} & (\beta^{\tau-1})^{-1} + U_\tau V_\tau^{-1} U_\tau^T \end{bmatrix} \tag{54}$$

where

$$U_\tau = -(\beta^{\tau-1})^{-1} \begin{bmatrix} s_2 \\ 0 \end{bmatrix} = - \begin{bmatrix} V_{\tau-1}^{-1} s_2 \\ U_{\tau-1} V_{\tau-1}^{-1} s_2 \end{bmatrix} \tag{55}$$

$$V_\tau = s_1 + C^T R^{-1} C - \begin{bmatrix} s_2^T & 0 \end{bmatrix} (\beta^{\tau-1})^{-1} \begin{bmatrix} s_2 \\ 0 \end{bmatrix} \tag{56}$$

$$= s_1 + C^T R^{-1} C - s_2^T V_{\tau-1}^{-1} s_2 \tag{57}$$

with initial conditions $U_2 = -(\beta^1)^{-1} s_2$ and $V_2 = s_1 + C^T R^{-1} C - s_2^T (\beta^1)^{-1} s_2$ and $\beta^1 = Q + C^T R^{-1} C$. The only matrices requiring inversion are V_τ and β^1, both $n \times n$ matrices.

5.3 Determinant Term

We will now derive a recursive equation to compute the determinant of Φ_2^τ. Taking the determinant of the block matrix in (36),

$$\log |\Phi_2^\tau| = \log \left|\Phi_2^{\tau-1}\right| + \log \left|\phi_{2yy} - \phi_{2yY} (\Phi_2^{\tau-1})^{-1} \phi_{2Yy}\right| \tag{58}$$

Looking at the update term, we have

$$\log\left|\phi_{2yy} - \phi_{2yY}(\Phi_2^{\tau-1})^{-1}\phi_{2Yy}\right| = \log\left|\hat{\Phi}_2\right| \quad (59)$$

$$= \log\left|C_2\Gamma_2 C_2^T + R_2\right| \quad (60)$$

The determinant of Φ_1^τ can be computed in a similar manner. As is, the update term requires the computation of the determinant of a $m \times m$ matrix, which can still be a daunting task. Under the assumption that the image noise is iid, the amount of computation reduces further.

5.3.1 Evaluation of Determinant with iid image noise

We will assume that the noise of the image is iid, i.e. $R = \sigma^2 I$, and that the covariance matrix is of the form $\Phi = C\Sigma C^T + \sigma^2 I$, with $\Sigma \in \mathbb{R}^{n\times n}$, $C \in \mathbb{R}^{m\times n}$ and $C^T C = I$. Let C' be a $m \times m$ orthonormal matrix such that $C' = [C\ X]$, where X is the matrix of the remaining orthonormal basis vectors. Let $\Sigma' = \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix}$, then we have

$$\Phi = C\Sigma C^T + \sigma^2 I \quad (61)$$

$$= C'\Sigma' C'^T + \sigma^2 I \quad (62)$$

We will now calculate the determinant of (62) by simultaneously diagonalizing the two terms of the sum. Let $\Sigma' = V\Lambda V^T$ be the eigen-decomposition of Σ', i.e. V is the matrix of eigenvectors where $V^T V = I$, and $\Lambda = \mathrm{diag}(\lambda_1, ..., \lambda_{n\tau}, 0, ..., 0)$ is the diagonal matrix of eigenvalues, where λ_i are the eigenvalues of Σ.

$$C'\Sigma' C'^T + \sigma^2 I = C'V\Lambda V^T C'^T + \sigma^2 I \quad (63)$$

Let $A = C'V$, and premultiply the RHS by A^T and postmultiply by A,

$$A^T(C'V\Lambda V^T C'^T + \sigma^2 I)A = \Lambda + \sigma^2 I \quad (64)$$

$$\log\left|A^T\Phi A\right| = \log\left|\Lambda + \sigma^2 I\right| \quad (65)$$

$$\log|\Phi| = \log\left|\sigma^2\left(\frac{1}{\sigma^2}\Lambda + I\right)\right| \quad (66)$$

$$= \log\left|\frac{1}{\sigma^2}\Lambda + I\right| + m\log\sigma^2 \quad (67)$$

Where in the third line we have used the fact that $|A| = 1$ because A is orthonormal. Noting that the determinant of a diagonal matrix is the product of the diagonal,

$$\log|\Phi| = \sum_{i=1}^{n}\log\left(\frac{\lambda_i}{\sigma^2} + 1\right) + m\log\sigma^2 \quad (68)$$

Thus the problem of finding the determinant of a $m \times m$ covariance matrix is reduced to that of finding the n eigenvalues of Σ when the R is an iid covariance matrix.

5.4 Trace Term

The trace term of the KL divergence can be reduced by using the matrix inversion lemma and some simple manipulation.

$$\begin{aligned}
\mathrm{tr}\left[(\Phi_2^\tau)^{-1}\Phi_1^\tau\right] &= \mathrm{tr}\left[\mathbf{R}_2^{-1} - \mathbf{R}_2^{-1}\mathbf{C}_2((\Sigma_2^\tau)^{-1} + \mathbf{C}_2^T\mathbf{R}_2^{-1}\mathbf{C}_2)^{-1}\mathbf{C}_2^T\mathbf{R}_2^{-1}\right]\Phi_1^\tau \quad (69)\\
&= \mathrm{tr}[\mathbf{R}_2^{-1}\Phi_1^\tau] - \mathrm{tr}\left[(\beta_2^\tau)^{-1}\mathbf{C}_2^T\mathbf{R}_2^{-1}\Phi_1^\tau\mathbf{R}_2^{-1}\mathbf{C}_2\right] \quad (70)\\
&= \mathrm{tr}[\mathbf{R}_2^{-1}(\mathbf{C}_1\Sigma_1^\tau\mathbf{C}_1^T + \mathbf{R_1})] \quad (71)\\
&\quad - \mathrm{tr}[(\beta_2^\tau)^{-1}\mathbf{C}_2^T\mathbf{R}_2^{-1}(\mathbf{C}_1\Sigma_1^\tau\mathbf{C}_1^T + \mathbf{R}_1)\mathbf{R}_2^{-1}\mathbf{C}_2]
\end{aligned}$$

Finally, the trace term becomes,

$$\begin{aligned}
\mathrm{tr}\left[(\Phi_2^\tau)^{-1}\Phi_1^\tau\right] &= \mathrm{tr}[\Sigma_1^\tau(\mathbf{C}_1^T\mathbf{R}_2^{-1}\mathbf{C}_1)] + \mathrm{tr}[\mathbf{R}_2^{-1}\mathbf{R}_1] \quad (72)\\
&\quad - \mathrm{tr}[(\beta_2^\tau)^{-1}(\mathbf{C}_2^T\mathbf{R}_2^{-1}\mathbf{R}_1\mathbf{R}_2^{-1}\mathbf{C}_2)]\\
&\quad - \mathrm{tr}[(\beta_2^\tau)^{-1}(\mathbf{C}_2^T\mathbf{R}_2^{-1}\mathbf{C}_1)\Sigma_1^\tau(\mathbf{C}_1^T\mathbf{R}_2^{-1}\mathbf{C}_2)]
\end{aligned}$$

The first three terms of the trace can be computed recursively. Let,

$$\alpha_\tau = \mathrm{tr}[\Sigma_1^\tau(\mathbf{C}_1^T\mathbf{R}_2^{-1}\mathbf{C}_1)] + \mathrm{tr}[\mathbf{R}_2^{-1}\mathbf{R}_1] - \mathrm{tr}[(\beta_2^\tau)^{-1}(\mathbf{C}_2^T\mathbf{R}_2^{-1}\mathbf{R}_1\mathbf{R}_2^{-1}\mathbf{C}_2)] \quad (73)$$

Then, the recursion is,

$$\begin{aligned}
\alpha_\tau &= \mathrm{tr}[\Sigma_{1xx}(C_1^TR_2^{-1}C_1)] + \mathrm{tr}[R_2^{-1}R_1] - \mathrm{tr}[V_\tau^{-1}(C_2^TR_2^{-1}R_1R_2^{-1}C_2)] \quad (74)\\
&\quad - \mathrm{tr}[V_\tau^{-1}U_\tau^T(C_2^TR_2^{-1}R_1R_2^{-1}C_2)U_\tau] + \alpha_{\tau-1}
\end{aligned}$$

The trace term is then computed as

$$\mathrm{tr}\left[(\Phi_2^\tau)^{-1}\Phi_1\right] = \alpha_\tau - \mathrm{tr}[(\beta_2^\tau)^{-1}\Psi^\tau] \quad (75)$$

where,

$$\Psi^\tau = \begin{bmatrix} \Psi^{\tau-1} & (\mathbf{C}_2^T\mathbf{R}_2^{-1}\mathbf{C}_1)\Sigma_{1Xx}(C_1^TR_2^{-1}C_2) \\ (C_2^TR_2^{-1}C_1)\Sigma_{1xX}(\mathbf{C}_1^T\mathbf{R}_2^{-1}\mathbf{C}_2) & (C_2^TR_2^{-1}C_1)\Sigma_{1xx}(C_1^TR_2^{-1}C_2) \end{bmatrix} \quad (76)$$

Note that β_2^τ and Ψ^τ are symmetric matrices with the same size, thus the trace of their product is simply the sum of the entries of the Hadamard product.

5.5 KL Divergence with iid Noise Assumption

If the image noise can be modeled as a iid Gaussian, i.e $R_1 = \sigma_1^2 I$ and $R_2 = \sigma_2^2 I$, some of the terms in the KL calculation simplify because C_1 and C_2 are matrices of orthonormal vectors. Specifically, the simplified equations are

$$z = \frac{1}{\sigma_2^2} C_2 \Sigma_{2xX} \Delta_2 (\mathbf{C}_2^T \mathbf{C}_1 \mu_1^{\tau-1} - \mu_2^{\tau-1}) - C_1 \mu_{1x} + C_2 \mu_{2x} \tag{77}$$

$$\beta_2^\tau = (\Sigma_2^\tau)^{-1} + \frac{1}{\sigma_2^2} I \tag{78}$$

$$\Delta_2 = I - \frac{1}{\sigma_2^2} (\beta_2^\tau)^{-1} \tag{79}$$

$$\Gamma_2 = \Sigma_{2xx} - \frac{1}{\sigma_2^2} \Sigma_{2xX} \Delta_2 \Sigma_{2Xx} \tag{80}$$

$$\alpha_\tau = \frac{1}{\sigma_2^2} \mathrm{tr}[\Sigma_{1xx}] + m \frac{\sigma_1^2}{\sigma_2^2} - \frac{\sigma_1^2}{\sigma_2^4} \mathrm{tr}[V_\tau^{-1}] - \frac{\sigma_1^2}{\sigma_2^4} \mathrm{tr}[V_\tau^{-1} U_\tau^T U_\tau] + \alpha_{\tau-1} \tag{81}$$

$$\Psi_\tau = \begin{bmatrix} \Psi_{\tau-1} & \frac{1}{\sigma_2^4} (\mathbf{C}_2^T \mathbf{C}_1) \Sigma_{1Xx} (C_1^T C_2) \\ \frac{1}{\sigma_2^4} (C_2^T C_1) \Sigma_{1xX} (\mathbf{C}_1^T \mathbf{C}_2) & \frac{1}{\sigma_2^4} (C_2^T C_1) \Sigma_{1xx} (C_1^T C_2) \end{bmatrix} \tag{82}$$

$$V_\tau = s_1 + \frac{1}{\sigma_2^2} I - s_2^T V_{\tau-1}^{-1} s_2 \tag{83}$$

A Appendix - Useful Matrix Identities

Matrix Inversion Lemma

$$(A^{-1} + VC^{-1}V^H)^{-1} = A - AV(C + V^H AV)^{-1} V^H A \tag{84}$$

Block Matrix Determinant

$$\begin{vmatrix} A & B \\ C & D \end{vmatrix} = |A| \left| D - CA^{-1}B \right| \tag{85}$$

Block matrix inversion

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} 0 & 0 \\ 0 & D^{-1} \end{bmatrix} + \begin{bmatrix} I \\ -D^{-1}C \end{bmatrix} Q^{-1} \begin{bmatrix} I & -BD^{-1} \end{bmatrix} \tag{86}$$

$$Q = A - BD^{-1}C \tag{87}$$

Mahalanobis Distance

$$M = \begin{bmatrix} A & B \\ B^T & D \end{bmatrix} \tag{88}$$

$$P = D - B^T A^{-1} B \tag{89}$$

$$z = \begin{bmatrix} x \\ y \end{bmatrix} \tag{90}$$

$$\|z\|_M^2 = \|x\|_A^2 + \left\|B^T A^{-1} x - y\right\|_P^2 \tag{91}$$

References

[1] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto. Dynamic textures. *International Journal of Computer Vision*, (2):91–109, 2003.

[2] S. M. Kay *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice-Hall Signal Processing Series, 1993.

[3] S. Kullback. *Information Theory and Statistics*. Dover Publications, New York, 1968.

[4] T. Cover and J. Thomas. *Elements of Information Theory*. John Wiley, 1991.

SVCL-TR 2004/02 rev4

November 2004

Efficient Computation of the KL Divergence between Dynamic Textures

Antoni B. Chan