A Hierarchical Wavelet-Based Image Model for Pattern Analysis and Synthesis

Clayton Scott and Robert Nowak
Dept. of Electrical and Computer Engineering, Rice University
6100 S. Main, Houston, TX, 77005

ABSTRACT
Despite their success in other areas of statistical signal processing, current wavelet-based image models are inadequate for modeling patterns in images, due to the presence of unknown transformations (e.g., translation, rotation, scaling) inherent in most pattern observations. In this paper we introduce a hierarchical wavelet-based framework for modeling patterns in digital images. This framework takes advantage of the efficient image representations afforded by wavelets, while accounting for unknown pattern transformations. Given a trained model, we can use this framework to synthesize pattern observations. If the model parameters are unknown, we can infer them from labeled training data using TEMPLAR (Template Learning from Atomic Representations), a novel template learning algorithm with linear complexity. TEMPLAR employs minimum description length (MDL) complexity regularization to learn a template with a sparse representation in the wavelet domain. We illustrate template learning with examples, and discuss how TEMPLAR applies to pattern classification and denoising from multiple, unaligned observations.

Keywords: Wavelets, pattern analysis, MDL

1. INTRODUCTION
Wavelet decompositions often provide very parsimonious image and pattern representations, and this feature has been exploited to devise powerful compression, denoising and estimation methods. Although wavelets provide sparse representations for many real world images, it is difficult to develop statistical models for patterns based directly on wavelet coefficients. This is because in many (perhaps most) applications, the pattern of interest undergoes an unknown or random transformation during data acquisition (e.g., variations in illumination, orientation, translation, and perspective). Modeling the wavelet expansion of such transformed data leads to distorted components, or even components that model the transformations instead of the structure of the underlying object or pattern.

The objective of this work is to develop a wavelet-based framework for modeling pattern observations that have undergone random transformations in the observation process. We introduce TEMPLAR (Template Learning from Atomic Representations), an algorithm that combines the edge-detection property of wavelets with minimum description length (MDL) complexity-regularization to automatically learn a low-dimensional pattern template from noisy, randomly transformed observations. Our approach is similar in spirit to that of Frey and Jojic; although in that work, the dimension of the template is fixed in advance and the basis vectors are adaptive. We work with a fixed wavelet basis, and allow the dimension of the template to vary.

In Section 2 we introduce our hierarchical framework and statistical model for describing patterns. In Section 3 we present TEMPLAR, an iterative algorithm for learning a pattern template from training observations. In Sections 4 and 5 we illustrate pattern learning and synthesis with two examples, and consider other applications of TEMPLAR and the hierarchical pattern framework.

2. HIERARCHICAL FRAMEWORK AND STATISTICAL MODEL
When a pattern is observed in an image, it can appear at any number of locations, orientations, scales, etc., in the image, depending on the spatial relationship between the image forming device and the pattern. Further uncertainty in pattern observations can be caused by lighting sources, background clutter, observation noise, and deformations of the pattern itself (if the pattern is not rigid, like a human face). We model these uncertainties in pattern observations with a hierarchical framework, based on the notion of deformable templates. For our purposes, a template is a

Email: cscott@rice.edu and nowak@rice.edu. Web: www.dsp.rice.edu
noise-free observation of a pattern that can be transformed into an arbitrary observation of the same pattern by applying a deformation to the template, and adding observation noise. Examples of deformations include global operations such as translations and rotations, as well as transformations with localized support, to represent local perturbations of the pattern.

Our hierarchical framework attempts to de-couple three different aspects of the pattern observations: the observation noise, the unknown transformations, and the template itself. Thus, a pattern observation is synthesized by taking the template, applying a randomly selected transformation, and adding a realization of the observation noise. The novelty of our approach is that we model the template in the wavelet domain. As we show in Section 3, this allows us to develop a template learning algorithm that takes advantage of the properties of the wavelet transform. We now describe in more detail the individual stages of the hierarchical framework.

We model template deformations with a finite set of linear transformations $\mathbf{\Gamma}_1, \ldots, \mathbf{\Gamma}_L$. Restricting to linear transformations still allows for a wide range of transformations, such as translation, rotation, scaling, and shearing. Furthermore, the aforementioned transforms all have sparse matrix representations, i.e., if they are operating on images with $N$ pixels, their matrix representations have $O(N)$ non-zero entries. This allows for linear-time computation and matrix inversion. In addition to linearity, we will also assume the transformation operators are invertible and, moreover, orthogonal. Of the transformations listed above, only translation is truly orthogonal when implemented on a computer, but the other transformations are “nearly” orthogonal in the following sense: if $\mathbf{\Gamma}_i$ is a matrix representation of the operator, then $\mathbf{\Gamma}_i^T \mathbf{\Gamma}_i$ differs from the identity matrix by $O(N)$ entries. This assumption will be most important when calculating the effect of various transformations on covariance matrices.

These transformations fixed before any analysis is done, and are chosen to cover (or at least reasonably approximate) the suspected range of possible transformations. This list may be reduced if we assume each training image has been crudely preprocessed to compensate for very gross transformations such as large translations. To limit the complexity of our model, we do not directly account for local pattern deformations. However, we may still capture variability in the pattern itself by treating the template as a random variable. In particular, we model the statistics of the template in the wavelet domain.

Let the random vector $\mathbf{W} = (W_1, \ldots, W_N)^T$ denote the wavelet coefficients (as computed by some orthogonal wavelet transform $\mathcal{W}$) of the pattern template, where $N$ is the total number of pixels in a pattern observation. In real-world images, we observe two “flavors” of wavelet coefficients: large coefficients, corresponding to wavelets that match edges in the image, and small coefficients, corresponding to smooth regions of the image or noise. We refer to the former as significant coefficients, and the latter as insignificant coefficients. We assign to each $W_i$ a hidden state variable $s_i$ taking on the values 0 or 1, with $s_i = 1$ indicating a significant coefficient, and $s_i = 0$ indicating an insignificant coefficient. This dichotomy is captured as follows: we model the marginal pdf of the insignificant wavelet coefficients as $p_{W_i|s_i}(w_i|s_i = 0) = N(w_i|\mu_{i,0}, \sigma_{i,0}^2)$; and, we model the marginal pdf of the significant wavelet coefficients as $p_{W_i|s_i}(w_i|s_i = 1) = N(w_i|\mu_{i,1}, \sigma_{i,1}^2)$. Here, $N(x|\mu, \sigma^2)$ denotes a Gaussian density with mean $\mu$ and variance $\sigma^2$, as a function of the variable $x$. For $m = 0, 1$, define $\mu_m \equiv (\mu_{1,m}, \ldots, \mu_{N,m})^T$, and define $\Sigma_m = \text{diag}(\sigma_{1,m}^2, \ldots, \sigma_{N,m}^2)$, where diag($a_1, \ldots, a_n$) denotes the $n \times n$ diagonal matrix with diagonal entries $a_1, \ldots, a_n$. To simplify our model, we assume $\mu_{i,0} = 0$ and $\sigma_{i,0}^2 = \sigma_{i,1}^2$ for each $i$. In other words, all insignificant wavelet coefficients are zero mean and have a common variance $\sigma_{i,0}^2$. This reflects our belief that the insignificant wavelet coefficients model smooth regions or noise. We collect the means and variances of the significant and insignificant wavelet coefficients together in the parameter vector $\boldsymbol{\theta} = \{\mu_1, \Sigma_1, \mu_0, \Sigma_0\}$. Because of our simplifying assumptions, we may also write $\theta = \{\mu_1, \Sigma_1, \mu_0, \Sigma_0\}.$

We assume that the wavelet coefficients are statistically independent. (This is a reasonable assumption because the wavelet basis functions are orthogonal and spatially localized. There are in fact dependencies between wavelet coefficients such as spatial clustering and persistence across scale which we do not treat in this paper.) Therefore the joint distribution of the wavelet coefficients of the pattern template is specified by $\theta$, together with a configuration $s = (s_1, \ldots, s_N)^T$ of state variables. To synthesize a pattern observation from this template, we first generate a realization $\mathbf{w}$ of wavelet coefficients according to the joint pdf $p_{W|\theta}(\mathbf{w}|\theta) = \prod_i p_{W_i|s_i}(w_i|s_i)$. Next, we obtain a realization $\mathbf{z} = \mathcal{W}^{-1}\mathbf{w}$ of the template by transforming $\mathbf{w}$ into the spatial domain. A transformation $\mathbf{\Gamma}_t$ is then selected from the list $\mathbf{\Gamma}_1, \ldots, \mathbf{\Gamma}_L$ according to some prior distribution, and applied to the spatial template to form a transformed pattern $\mathbf{y} = \mathbf{\Gamma}_t \mathbf{z}$. Finally, the observed image $\mathbf{x}$ is created by corrupting $\mathbf{y}$ with additive observation noise, which we model as zero-mean iid Gaussian with variance $\sigma^2_{\text{obs}}$. The conditional density of the observed image
is given by

\[ p(x \mid \theta, s, \ell) = \mathcal{N}(x \mid \Gamma^T \mu, \Gamma^T \Sigma \Gamma^T + \sigma^2_n I), \]

where \( \mu \equiv (\mu_1, s_1, \ldots, \mu_N, s_N)^T \) and \( \Sigma \equiv \text{diag}(\sigma^2_1, \ldots, \sigma^2_N) \). Since \( \Gamma \) and \( \mathcal{W} \) are orthogonal operators, we have \( \sigma^2_n I = \Gamma \mathcal{W}^T (\sigma^2_n I) \mathcal{W} \Gamma^T \), and therefore we may write the covariance matrix of the above density as \( \Gamma \mathcal{W}^T (\Sigma + \sigma^2_n I) \mathcal{W} \Gamma^T \). In light of this fact, we see that it is not necessary to model the observation noise separate from the wavelet coefficients. Henceforth we assume any observation noise is captured by the wavelet domain statistical model, and that the density on the space of observed images is given by

\[ p(x \mid \theta, s, \ell) = \mathcal{N}(x \mid \Gamma^T \mu, \Gamma^T \Sigma \Gamma^T + \sigma^2_n I). \]

3. COMPLEXITY REGULARIZED TEMPLATE LEARNING VIA TEMPLAR

In this section we introduce a method for learning template parameters \( \theta \) and \( s \) from training data, i.e. a collection \( \mathcal{X} = (x^1, \ldots, x^L) \) of statistically independent observations of the same underlying pattern. In the process, we also learn the indices \( \mathcal{L} = (\ell_1, \ldots, \ell_T) \) of the transformations giving rise to each observation. Our approach is to perform penalized maximum likelihood (PML) estimation of the parameters \( \theta, s \), and \( \mathcal{L} \). In particular, we would like to maximize the objective function

\[ F(\theta, s, \mathcal{L}) = \log p(\mathcal{X} \mid \theta, s, \mathcal{L}) + c(s) \]

where \( c(s) \) is a complexity penalty term that decreases as the number of significant coefficients increases. We include \( c(s) \) to balance the trade-off between fitting the data and model complexity, and thus to promote the learning of a template with a sparse wavelet representation.

We select a minimum description length (MDL) based penalty term of the form

\[ c(s) = -2k \log(N), \]

where \( k = \sum s_i \) is the total number of significant coefficients. The negative of this quantity is interpreted as the number of bits required to encode the location and values of the means and variances of the significant coefficients. In detail, following a derivation similar to that of Saito and Coifman,5 each significant coefficient requires approximately \( \log_2(N) \) bits to encode its location, \( \frac{1}{2} \log_2(N) \) bits to encode its mean, and \( \frac{1}{4} \log_2(N) \) bits to encode its variance. Since the negative log-likelihood is the Shannon code length required to encode the data, we see that maximizing \( F \) is equivalent to minimizing the total code (description) length required to encode both data and model parameters.

Joint maximization of \( F \) over all three parameters is intractable. Our approach is to find a (possibly local) solution to this optimization problem with TEMPLAR (Template Learning from Atomic Representations), an iterative alternating-maximization algorithm. The premise is to maximize over one parameter at a time, while holding the other two fixed. We first initialize estimates \( \hat{\theta}_0 \) and \( \hat{\mathcal{L}}_0 \) of the states and hidden transformations, by stipulating, for example, that all wavelet coefficients are insignificant, and all transformations are the identity transformation. TEMPLAR then proceeds according to

\[ \hat{\theta}_j = \arg \max_{\theta} F(\theta, \hat{s}_{j-1}, \hat{\mathcal{L}}_{j-1}) \]

\[ \hat{s}_j = \arg \max_{s} F(\hat{\theta}_j, s, \hat{\mathcal{L}}_{j-1}) \]

\[ \hat{\mathcal{L}}_j = \arg \max_{\mathcal{L}} F(\hat{\theta}_j, \hat{s}_j, \mathcal{L}) \]

Since each step involves a maximization, this produces a non-decreasing sequence of penalized log-likelihood values. The process continues until \( F(\hat{\theta}_j, \hat{s}_j, \hat{\mathcal{L}}_j) = F(\hat{\theta}_{j-1}, \hat{s}_{j-1}, \hat{\mathcal{L}}_{j-1}) \), which is guaranteed to happen in a finite number of iterations, as proved in Theorem 1 below. When the algorithm stops, the current values of \( \hat{\theta}_j \) and \( \hat{s}_j \) are estimates for the template, and the current value of \( \hat{\mathcal{L}}_j \) contains the best estimate of the transformations that generated the training images from the learned template.

Each step of TEMPLAR is simple and efficient. One iteration requires \( O(NLT) \) operations, where \( N \) is the total number of pixels in a pattern observation, \( L \) is the number of transformations, and \( T \) is the number of training images. The use of complexity-regularization combined with the inherent edge detection capabilities of wavelets
produces a low-dimensional template that captures the defining features of the pattern of interest, but does not represent background clutter or noise. The details of the algorithm are given in Appendix A.

For notational convenience, we define \( \alpha_j \equiv (\theta_j, s_j, L_j) \) and \( F_j \equiv F(\alpha_j) \). As noted earlier, the sequence \( \{F_j\} \) is non-decreasing. Note that the particular sequence (as well as the terminal value of \( \alpha \)) depends on the initial settings \( s_0 \) and \( L_0 \).

**Theorem 1.** The sequence of penalized log-likelihood values \( \{F_j\} \) generated by TEMPLAR (according to Equations (2-4)) converges and reaches its limit in a finite number of iterations.

**Proof.** First observe that there are \( 2^N \) possible configurations of the state vector \( s \), and \( L_T \) possible configurations of \( L \). Furthermore, each \( \theta_j \) is determined by \( X, s_{j-1} \), and \( L_{j-1} \). Thus, there are \( 2^N L_T \) different possibilities for \( \alpha_j \), and therefore there are at most \( 2^NL_T \) different values for \( F_j \). A non-decreasing sequence that takes on a finite number of values must converge and reach its limit after finitely many terms in the sequence. \( \Box \)

It follows that TEMPLAR terminates in a finite number of iterations. However, Theorem 1 does not tell us about the convergence of the sequence \( \{\alpha_j\} \) of parameter estimates, which is what we are ultimately interested in. The following result allows us to address this issue:

**Theorem 2.** If \( F_j = F_{j-1} \), then with probability one, we have \( \alpha_j = \alpha_{j-1} \).

The proof is found in an appendix. Here “with probability one” refers to the training images \( x^1, \ldots, x^T \), which are randomly drawn from \( \mathbf{R}^{N_T} \) according to a Gaussian distribution.

This Theorem tells us that by the time TEMPLAR stops, the sequence \( \{\alpha_j\} \) of parameter estimates has converged (with probability one). To see this, suppose the algorithm stops after \( j^* \) iterations, i.e. \( F_{j^*} = F_{j^*-1} \). By Theorem 2, \( \alpha_{j^*} = \alpha_{j^*-1} \), with probability one. But \( \alpha_{j^*+1} \) is determined from \( \alpha_{j^*} \) by Equations (2)-(4) just as \( \alpha_{j} \) is determined from \( \alpha_{j-1} \). Therefore, \( \alpha_{j^*+1} = \alpha_{j^*} \), and by induction, \( \alpha_j = \alpha_{j^*} \) for all \( j > j^* \). In other words, \( \alpha_{j^*} \) is (with probability one) the limit of the sequence \( \{\alpha_j\} \). We have thus proved

**Corollary 1.** The final estimate of the template and hidden transformations produced by TEMPLAR is the limit of the sequence \( \{\alpha_j\} \), with probability one.

This result eliminates the possibility of “cycling,” i.e. having two or more different configurations of \( \theta, s \), and \( L \) that yield the same penalized log-likelihood value.

### 4. Examples of Template Learning and Pattern Synthesis

We illustrate template learning and pattern synthesis with two examples. In the first example, we apply TEMPLAR to twenty observations of a randomly translated and rotated toy airplane (see Figure 1 (a)). These \( 128 \times 128 = 16384 \) dimensional images were obtained with a digital camera. Note the varying background, lighting intensity, and location of lighting source. Using the Haar wavelet, TEMPLAR converges to a 853 dimensional template in seven iterations. In the second example, we apply TEMPLAR to twenty observations of a randomly translated face with different facial expressions. The images in this experiment are \( 128 \times 128 = 16,384 \) dimensional. Using the Daubechies 6 wavelet, TEMPLAR converges to a 543 dimensional template in seven iterations. For both examples, the template is low dimensional and represents the defining structure of the pattern, but does not represent noise or the background. Having learned these two templates, we used them to synthesize observations of each pattern. In both cases, these look like the training images. Notice that in the second example, the random nature of the template is able to reproduce some of the facial expressions, even though translations were the only template deformations explicitly modeled. In future work we hope to model statistical dependencies between significant wavelet coefficients to more accurately reproduce facial expressions. Figures 1 and 2 show the results of these experiments.

### 5. Other Applications of TEMPLAR

We mention briefly two other applications of TEMPLAR and our hierarchical pattern framework: pattern classification and denoising from multiple observations. If we are given training images for two or more classes of patterns, we can apply TEMPLAR to produce templates for each class, and use the resulting pattern models for likelihood-based
classification. Specifically, if we have trained models $\{\theta_c, s_c\}_{c=1}^{C}$ for $C$ different classes of patterns, we can use these models to classify an unlabeled test image $x$ according to a generalized likelihood ratio test (GLRT):

$$c^* = \arg \max_c \left[ \max_{\ell} p(x | \theta_c, s_c, \ell) \right].$$  

(5)

The GLRT selects the class that has the highest likelihood when evaluated using the most likely transformation for that class. This results in a classifier that is independent of the transformation $\Gamma_\ell$ that gave rise to $x$. We may also choose to evaluate the likelihood in Equation (5) using only the significant template coefficients. Since the significant coefficients only model pattern structure and not background objects, this gives us a low dimensional subspace classifier that is invariant to clutter. The number of operations required to classify an image is $O(NM)$.

The second application is denoising from multiple observations of the same pattern. Previous work on this problem assumes that the pattern is aligned in each of the observations.6 With our framework, we no longer require this assumption. TEMPLAR automatically registers the observations, and performs denoising by averaging and then setting some wavelet coefficients to zero (although not according to a thresholding rule). The coefficients that are set to zero are precisely the insignificant coefficients.

6. CONCLUSION

We present a wavelet-based approach for modeling observations of patterns that undergo unknown transformations. We introduce TEMPLAR, an algorithm that combines the compression property of wavelets with MDL complexity-regularization to automatically learn a low-dimensional template from training data. The dimension of the template is automatically inferred from the data. This is in contrast to previous work that either assumes the observations are already aligned,6 or that fix the dimension of the template in advance.2 Once the template has been learned, it can be used to synthesize observations that look like the training images. TEMPLAR and the hierarchical pattern framework can also be used for pattern classification and denoising from multiple, randomly transformed observations of the same pattern.7

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation, grant no. MIP-9701692, the Army Research Office, grant no. DAAD19-99-1-0349, and the Office of Naval Research, grant no. N00014-00-1-0390.

APPENDIX A. THE DETAILS OF TEMPLAR

In this appendix we provide explicit solutions to the optimization problems in Equations (2-4) and give the associated computational complexities.

A.1. Estimating Gaussian Mixture Means and Variances

Consider the problem of maximizing $F$ over the parameter $\theta = \{\mu, \Sigma, \sigma^2\}$, given training data $X = (x^1, \ldots, x^T)$ and estimates $\hat{s}$ and $\hat{L}$ for the states and hidden transformations. The penalty term $c(s)$ does not depend on $\theta$, so this is equivalent to maximizing the likelihood $p(\mathcal{X} | \theta, s, L)$. where $p(x^t | \theta, s, \ell_t)$ is given by Equation 1. Denote $\mathcal{W} \Gamma^{-1} x^t$ by $w^t = (w^t_1, \ldots, w^t_N)^T$. By orthogonality of $\Gamma_\ell$ and $\mathcal{W}$, it follows from Equation (1) that

$$p(w^t | \theta, s, \ell_t) = \mathcal{N}(w^t | \mu, \Sigma).$$  

(6)

Recall that the i-th wavelet coefficient is distributed as $\mathcal{N}(\mu_{i,s}, \sigma^2_{i,s})$. Since the wavelet coefficients are independent and the training images are independent, Equation (6) implies

$$\hat{\mu}_{i,1} = \frac{1}{T} \sum_{t=1}^{T} w^t_{i}$$

$$\hat{\sigma}^2_{i,1} = \frac{1}{T} \sum_{t=1}^{T} (w^t_{i} - \hat{\mu}_{i,1})^2$$
are maximum likelihood estimates for the parameters of the Gaussian densities modeling the significant coefficients. If the number of training images $T$ is small, we may wish to normalize by $T-1$ instead of $T$ to achieve an unbiased estimate for the variances.

For the insignificant coefficients, which have zero mean, estimation of the variances would follow a similar formula, except that we are assuming all insignificant coefficients have the same variance $\sigma_0^2$. Therefore, we compute the common variance by averaging the maximum likelihood estimates of these variances over all coefficients currently in the insignificant state (as specified by $\bar{s}$):

$$\hat{\sigma}_0^2 = \frac{1}{N-k} \sum_{i=1}^{N} (1 - s_i) \frac{1}{T} \sum_{t=1}^{T} (w_t^i)^2$$

where $k = \sum s_i$, the number of significant wavelet coefficients. To avoid over fitting the model, if $\sigma_{i,1}^2 < \sigma_0^2$ for some $i$, we set $\sigma_{i,1}^2 = \sigma_0^2$.

Since $\mathcal{W}$ and $\mathbf{G}_{\ell}^{-1}$ can be computed in $O(N)$ operations, the process of estimating $\mathbf{\theta}$ requires $O(NT)$ operations.

### A.2. Determining States of Gaussian Mixtures

In order to maximize $F$ with respect to $s$, it is useful to write

$$\log p(X | \mathbf{\theta}, s, \mathcal{L}) = \sum_{i=1}^{N} f_i(s_i)$$

where $f_i$ is the contribution to the overall log-likelihood from the $i^{th}$ coefficient, as a function of the state variable $s_i$. The change of variables formula implies that

$$p(\mathcal{W}\mathbf{T}_{\ell}^{-1}X \mid \mathbf{\theta}, s, \mathcal{L}) = p(X \mid \mathbf{\theta}, s, \ell).$$

By orthogonality of $\mathcal{W}$ and $\mathbf{G}_\ell$, we have $|\mathcal{W}\mathbf{T}_{\ell}^{-1}| = 1$, so from Equation (6),

$$p(x^i \mid \mathbf{\theta}, s, \ell) = \mathcal{N}(w^i \mid \mu, \Sigma) = \frac{1}{\sqrt{2\pi |\Sigma|}} \exp \left\{-\frac{1}{2} (w^i - \mu)^T \Sigma^{-1} (w^i - \mu) \right\}. \tag{7}$$

Since the observations are assumed statistically independent,

$$p(X \mid \mathbf{\theta}, s, \mathcal{L}) = \prod_{t=1}^{T} p(x^t \mid \mathbf{\theta}, s, \ell_t)$$

Therefore we have

$$\log p(X \mid \mathbf{\theta}, s, \mathcal{L}) = \sum_{t=1}^{T} \log p(x^t \mid \mathbf{\theta}, s, \ell_t)$$

$$= \sum_{t=1}^{T} \left[ -\frac{1}{2} \log(2\pi |\Sigma|) - \frac{1}{2} \sum_{i=1}^{N} \frac{(w_t^i - \mu_{i,s_i})^2}{\sigma_{i,s_i}^2} \right]$$

$$= -\frac{1}{2} \sum_{t=1}^{T} \sum_{i=1}^{N} \left[ \frac{T}{N} \log 2\pi + \log \sigma_{i,s_i}^2 + \frac{(w_t^i - \mu_{i,s_i})^2}{\sigma_{i,s_i}^2} \right]$$

$$= -\frac{1}{2} \sum_{i=1}^{N} \left[ \sum_{t=1}^{T} \frac{T}{N} \log 2\pi + T \log \sigma_{i,s_i}^2 + \sum_{t=1}^{T} \frac{(w_t^i - \mu_{i,s_i})^2}{\sigma_{i,s_i}^2} \right].$$

In the third equality, we use the fact that $|\Sigma| = \prod_{i=1}^{N} \sigma_{i,s_i}^2$. Therefore we may define for $m = 0, 1$

$$f_i(m) = -\frac{1}{2} \left[ \frac{T}{N} \log 2\pi + T \log \sigma_{i,m}^2 + \sum_{t=1}^{T} \frac{(w_t^i - \mu_{i,m})^2}{\sigma_{i,m}^2} \right].$$
Now define \( d_i = f_i(1) - f_i(0) \). This is the increase in the log-likelihood we achieve when the \( i \)-th coefficient is significant as opposed to insignificant. Since our penalty on \( s \) is only a function of the number of significant coefficients, if we are going to have any significant coefficients, they should be the coefficients with the largest \( d_i \) values. Therefore we order the coefficients so that \( d_{i_1} \geq d_{i_2} \geq \cdots \geq d_{i_N} \). The maximum increase in the log-likelihood that can occur from having exactly \( k \) significant coefficients is \( \sum_{j=1}^k d_{i_j} \). But having \( k \) significant coefficients carries with it a penalty of \( c(s|k) \). Hence, the number of significant coefficients that maximizes the penalized log-likelihood is

\[
k^* = \arg\max_k \left[ c(s|k) + \sum_{j=1}^k d_{i_j} \right],
\]

and this maximum is attained by setting \( s_{ij} = 1 \) for \( j = 1, \ldots, k^* \), and \( s_{ij} = 0 \) for \( j = k^* + 1, \ldots, N \).

Note that the values \( w'_i \) used to compute \( d_i \) were already computed in the previous step, so they do not need to be computed again. The number of operations required to compute the \( d_i \) is \( O(NT) \), while the number of operations required to compute \( k^* \) is \( O(N \log N) \), due to the sorting and maximizing.

A.3. Inferring Hidden Transformations

As in Section A.1, the penalty \( c(s) \) does not involve \( L \), so PML estimation of \( L \) given \( \hat{\theta}_j, \hat{s}_j \) is equivalent to maximum likelihood estimation. Furthermore, since the pattern observations are statistically independent, we may estimate the hidden transformation for each training image independently. The ML estimate \( \hat{\ell}_i = (\hat{\ell}_1, \ldots, \hat{\ell}_T) \) is given by

\[
\hat{\ell}_i = \arg\max_\ell p(x' | \hat{\theta}_j, \hat{s}_j, \ell).
\]

We solve this optimization problem by computing the above likelihood for all \( L \) transformations. This likelihood can be computed from Equation (7), which requires \( O(N) \) operations. We repeat this computation \( L \) times for each of the \( T \) training images, so computing \( \hat{L} \) requires \( O(NLT) \) operations. This dominates the other two steps in TEMPLAR (assuming \( LT > \log N \), a very reasonable assumption), so the overall running time of TEMPLAR is \( O(NLT) \).

APPENDIX B. PROOF OF THEOREM 2

Suppose \( F_j = F_{j-1} \). If \( \alpha_j \neq \alpha_{j-1} \), then \( \theta_j \neq \theta_{j-1} \) or \( s_j \neq s_{j-1} \) or \( \mathcal{L}_j \neq \mathcal{L}_{j-1} \). The theorem will be proved if we show that each of these three cases occurs with probability zero. We will need the following two facts:

Remark 1. If \( F_j = F_{j-1} \), then

\[
\begin{align*}
F(\hat{\theta}_{j-1}, \hat{s}_{j-1}, \hat{\mathcal{L}}_{j-1}) & = F(\hat{\theta}_j, \hat{s}_{j-1}, \hat{\mathcal{L}}_{j-1}) \\
F(\hat{\theta}_j, \hat{s}_j, \hat{\mathcal{L}}_{j-1}) & = F(\hat{\theta}_j, \hat{s}_j, \hat{\mathcal{L}}_{j-1}) \\
F(\hat{\theta}_j, \hat{s}_j, \hat{\mathcal{L}}_{j-1}) & = F(\hat{\theta}_j, \hat{s}_j, \hat{\mathcal{L}}_{j})
\end{align*}
\]

This result follows immediately from the definition of TEMPLAR in Equations (2)-(4), since \( \theta_j, s_j, \) and \( \mathcal{L}_j \) are the solutions of maximizations, and therefore do not decrease the value of the penalized log-likelihood.

Remark 2. If \( p: \mathbb{R}^n \rightarrow \mathbb{R} \) is a nonzero polynomial in \( n \) variables, then the set of zeros of \( p \) has measure zero with respect to Lebesgue measure. See pp. 28-29 of \(^6\) for a proof.

I haven’t had time to write the rest of this up. It’s an easy application of the above two properties.

REFERENCES


Figure 1. Airplane pattern (a) 20 observations of the same airplane under different translations and rotations, and with different backgrounds and lighting conditions. (b) Starting in upper left corner and going clockwise: Mean value of learned template (transformed into the spatial domain) with 853 (out of 16384) significant coefficients; a synthesized observation; mean value of learned template using all 16,384 coefficients; map of significant wavelet coefficients.
Figure 2. Human face (a) Randomly translated observations with different facial expressions, used for training. (b) Mean value of learned template (transformed into the spatial domain) with 543 (out of 16,384) significant coefficients. (c) Synthesized observations.