

# TRANSLATING IMAGES BY UNSUPERVISED ESTIMATION OF SWITCHING FILTERS

Rómer Rosales, Kannan Achan, and Brendan Frey

Probabilistic and Statistical Inference Laboratory  
University of Toronto, Toronto, ON, Canada  
<http://www.psi.toronto.edu>

## ABSTRACT

We propose a method for altering pixel statistics of one image according to another (source) image. Given an input or observed image (probably degraded by one or more unknown processes), and a source image exhibiting the general patch (group of pixels) properties expected in the input image (before degradation), we seek to infer the original image and the process that affected it to produce the observed image. The foundation of our approach is to transform known image patches with desired statistics to patches found in the input image using a finite set of filters or transformations. These transformations are unknown; thus they also must be estimated. We cast this problem as an approximate probabilistic inference problem and show how it can be approached using belief propagation and expectation maximization. Experimental results for joint image restoration and filter estimation are presented.

## 1. INTRODUCTION AND RELATED WORK

We describe an approach for altering image properties (statistics) without modifying image *content*. The desired properties are specified by example, using a given *source* image, whose patches define an empirical patch probability distribution. The goal is to develop a general approach for transforming an *observed* image into another with preferable properties and at the same time for estimating the one or more transformations (*e.g.*, linear filters) that relate the observed and source image patches.

Many signal processing tasks are special cases of the above problem. Examples in image processing include: image de-noising, where we seek to remove *unwanted* noise from a given image to achieve a visually clearer one; and image super-resolution, where given a low-resolution image, the goal is to estimate a high-resolution version of that same image. More generally, the problem is to discover what the original latent image (signal) looked like before it underwent the effect of some unknown (or partially known) process. In this paper, we provide an approach for recovering the latent image and the unknown process, given the

observed image and example images that *roughly* exhibit the desired properties.

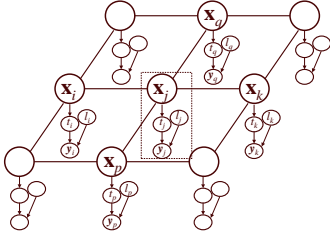
A large variety of methods have been proposed to approach specific related problems, *e.g.*, [1, 2, 3, 4, 5, 6]. The approach presented here is more closely connected to [3, 2, 1] in the sense that the joint distribution of the latent image (*i.e.*, image to be estimated) is represented as a Markov Random Field (MRF) with pairwise potentials. Our approach is also motivated by [4] from the computer graphics literature. However, there are critical differences. First, we use an unsupervised framework. In some cases it is possible to have several example pairs of degraded and original images. Using these examples, the unknown process could be estimated more easily (supervised learning). However, in many cases these example pairs do not exist. Previous work was based on supervised learning, we focus on the latter (unsupervised) problem. Second, it is sometimes assumed that the *degrading* process is known (*e.g.*, a blurring process). In this work, we do not assume we know this process and intend to uncover it. These two differences are up to some point related. Third, we derive new algorithms for estimating both the latent image and the transformations.

## 2. IMAGE TRANSLATION MODEL

### 2.1. Definitions and Setup

We will represent an image as a set of overlapping patches. Let  $\mathcal{Y}$  be the input or observed image, formed by a set of patches  $\mathbf{y}_p$ , with  $p \in \mathcal{P}$ ,  $\mathcal{P} = \{1, \dots, P\}$  and  $\mathbf{y}_p \in \mathbb{R}^S$ . Here,  $S$  is the number of pixels in each patch (this assumes one real value per pixel; however  $S$  could also account for representations using multiple channels or also filter responses instead of pixels). Consider also a latent image  $\mathcal{X}$ , with patches  $\mathbf{x}_p \in \mathbb{R}^T$ ;  $\mathcal{X}$  will be the image to be inferred or estimated. Let  $\mu$  denote a known image (or images) here referred to as the source or dictionary image. Assume that the set of patches in  $\mu$  are a representative sample which possesses the patch statistics that we wish  $\mathcal{X}$  to display.

Consider a set of patch transformations (*i.e.*, patch translators)  $\Lambda = \{\Lambda_l\}_{l=\{1, \dots, L\}}$ , where  $\Lambda_l : \mathbb{R}^S \rightarrow \mathbb{R}^T$ . In our



**Fig. 1.** Chain graph representing the model joint probability distribution.

model, the task of a single  $\Lambda_i$  is to transform a latent patch into an observed patch. These transformations are initially unknown, and we will try to estimate them. This loosely accounts for discovering the set of processes that altered the observed  $\mathcal{Y}$  (e.g., blurring) from its original version. Here we assume that this original image share the patch properties found in  $\mu$ . The image may have been altered by multiple processes, perhaps patch dependent. We will assume a finite number of processes. The random variable  $l_p$  will represent the index of the processes that transformed patch  $\mathbf{x}_p$  and  $\mathbf{l} = (l_1, \dots, l_P)$ .

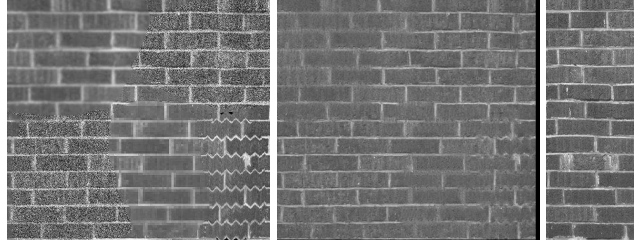
Consider the operator set  $\Gamma$ , each element  $\Gamma_k$  produces a patch when applied to an image. We will use this set to perform topological transformations of the patches in  $\mu$ . In this paper  $\Gamma_k$  is restricted to perform just patch extraction (i.e., obtain a square patch  $\mu_p \in \mathbb{R}^{T^2}$ ). This definition is used for generality, since in practice  $\Gamma_k$  could achieve other class of topological transformations such as rotation and shearing (not considered here). We use the random variable  $t$  to denote the  $t$ -th element of the set  $\Gamma$ , and  $\mathbf{t} = (t_1, \dots, t_P)$  to represent the topological transformations for all patches in the image.

## 2.2. Probabilistic Model

We defined our model to have joint probability distribution represented by the chain graph of Fig. 1, which can be factorized as the product of local conditional probabilities associated with the directed edges and positive potential functions associated with the undirected edges [7, 8]:

$$p(\mathcal{Y}, \mathcal{X}, \mathbf{l}, \mathbf{t} | \Gamma, \Theta, \mu) = \prod_{p \in \mathcal{P}} p(\mathbf{y}_p | l_p, t_p, \Gamma, \Theta, \mu) \prod_{p \in \mathcal{P}} P(t_p | \mathbf{x}_p) \prod_{p \in \mathcal{P}} P(l_p) \frac{1}{Z} \prod_{c \in \mathcal{C}} \psi_c(\mathbf{x}_{p \in c}), \quad (1)$$

with  $\{c \in \mathcal{C}\}$  denoting the set of latent image patches that belong to clique  $c$  in the MRF at the upper layer in Fig. 1,  $\mathcal{C}$  the set of cliques in the MRF sub-graph, and  $\psi_c$  the clique potential functions.



**Fig. 2.** Input ( $\mathcal{Y}$ ), inferred ( $\mathcal{X}$ ), and source ( $\mu$ ) images.

In this paper, every image patch  $\mathbf{y}_p$  follows a conditional Gaussian distribution given the patch transformation index  $l_p$  and the topological transformation  $t_p$ :

$$p(\mathbf{y}_p | l_p, t_p, \Gamma, \Theta, \mu) = \mathcal{N}(\mathbf{y}_p; \Lambda_{l_p}(\Gamma_{t_p} \mu), \Psi_p), \quad (2)$$

where  $\Theta = \{\Lambda, \Psi\}$  denotes the distribution parameters and  $\Psi = \{\Psi_p\}_{p \in \mathcal{P}}$ . We set  $\Lambda_i$  to be linear filters; however since the inferred patches are not the result of a linear function of the observed image patches, this is different than simply transforming the image patches linearly. Instead, by this we are imposing a constraint on the *flexibility* on each of the  $L$  transformations (it may be advantageous to also allow for non-linear filters).

We consider  $p(\mathbf{x})$  to be a pairwise Gaussian MRF, with potentials proportional to the inverse distance  $d$  between two patches;  $d$  is computed only on overlapping areas in the associated patches, in a way similar to [3].

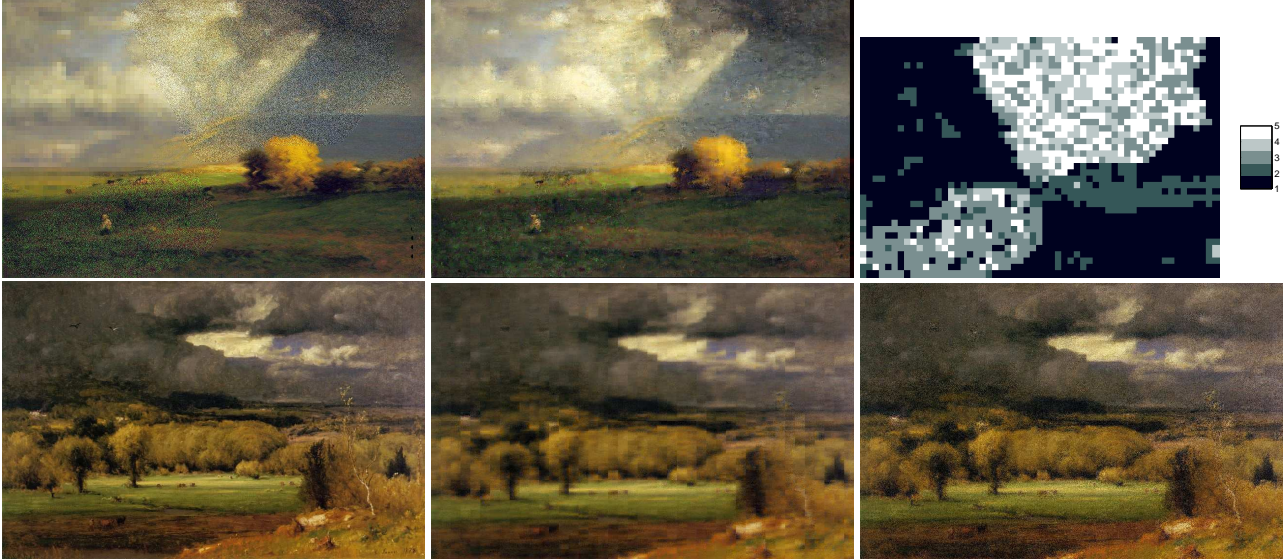
Finally, we link our discrete transformation random variable with the continuous latent random variable  $\mathcal{X}$  by:

$$p(t_p | \mathbf{x}_p) = \begin{cases} 1 & \text{if } \mathbf{x}_p = \Gamma_{t_p} \mu \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

A transformation  $t_p$  will have non zero conditional probability if its patch  $\mathbf{x}_p$  is equal to the patch taken from the dictionary  $\mu$  using transformation  $t_p$  (here we assume that the set  $\Gamma$  is such that it produces unique patches).

## 3. ALGORITHMS FOR LEARNING / INFERENCE

The chain graph in Fig. 1 contains an undirected sub-graph with loops; even if all the filter parameters  $\Theta$  were known, inference (computing the marginal conditional distribution over patches in  $\mathcal{X}$  given  $\mathcal{Y}$ ) is computationally intractable in general, with complexity  $\mathcal{O}(|\mathcal{K}|^{|\mathcal{P}|})$ , with  $|\mathcal{K}|$  the number of possible states that each patch  $\mathbf{x}_p$  can take. However, there exist approximation algorithms; one of them, based on alternating optimizations [9], is Expectation Maximization (EM). EM requires computing posterior distributions over  $\mathbf{l}$  and  $\mathbf{t}$ , that in turn requires computing conditional marginal distributions for each node of the undirected portion of our chain graph; as we have seen, this is computationally intractable. We divide this section into (1) inferring the latent



**Fig. 3.** Top: input image  $\mathcal{Y}$ , *Passing Clouds* by George Inness (left), inferred image (middle) and the corresponding (thresholded) *preferred* filter labels (right), note how the filter location matches with the different noise processes in the input image; bottom: one of the source images, *The Coming Storm* (left) and same image after two estimated filters (1,3) were applied (middle and right), corresponding to pixelation and correlated noise.

image (usually referred to as inference) and (2) estimating the model parameters (usually referred to as learning).

### 3.1. Inference and Approximate E-step

We assume that the reader has some familiarity with the EM algorithm (see *e.g.*, [10]). We view the E-step as equivalent to computing the posterior:

$$P(l_q, t_q | \mathcal{Y}) \propto \int_{\mathcal{X}} \sum_{\mathcal{I} \setminus l_q, t \setminus t_q} \prod_{p \in \mathcal{P}} p(\mathbf{y}_p | l_p, t_p) P(t_p | \mathbf{x}_p) p(\mathcal{X}) d\mathcal{X},$$

but we cannot solve this integral; thus we are forced to find an approximate way to perform the E-step.

**Approximating observed patch conditional distributions.** Let us say that we have some estimate for the filters  $\Lambda_l$  and  $\Psi_p$  (initially they may be random). For every  $p$ , we can select the set  $\mathcal{K}_p$  of  $K$  most likely topological transformations given the dictionary  $\mu$ . This can be easily done for each patch once  $P(t_p | \mathcal{Y})$  (see below) is computed for every topological transformation  $t_p$ . This approximation is necessary for computational reasons and can be made as exact as desired. This approximation was used in [3]; it accounts for cutting off the *tails* of the joint distribution  $P(t_p | \mathcal{Y})$ . However, it is still difficult to compute  $P(t_p | \mathcal{Y})$

**Inferring the latent image.** In computing  $P(t_p, l_p | \mathcal{Y})$  one key problem is that of inferring a posterior marginal distribution over  $\mathbf{x}_p$  (*i.e.*, marginalizing all the other patches  $\mathbf{x}_q$ ). One way to approximate this computation is by performing *loopy* belief propagation in the MRF for  $\mathcal{X}$  to com-

pute the posterior marginals over each  $\mathbf{x}_p$ . Loopy belief propagation accounts for approximating  $p(\mathbf{x}_p | \mathcal{Y})$  by using the belief propagation message passing updates  $m_{i \rightarrow j}$  [8] for several iterations, in our model written as follows:

$$m_{i \rightarrow j}(\mathbf{x}_j) = \sum_{\mathbf{x}_i = s_i} P(\mathbf{x}_i | t_i) \psi_{ij}(\mathbf{x}_i, \mathbf{x}_j) \prod_{k \in \mathcal{N}(i) \setminus j} m_{k \rightarrow i}(\mathbf{x}_i)$$

$$b_i(\mathbf{x}_i) = P(\mathbf{x}_i | t_i) \prod_{k \in \mathcal{N}(i)} m_{k \rightarrow i}(\mathbf{x}_i),$$

with  $\mathcal{N}(i)$  the neighbors and  $s_i$  the candidates for  $\mathbf{x}_i$ .

These updates guarantee that upon convergence the marginal probabilities  $p(\mathbf{x}_p | \mathcal{Y})$ , obtained by simply normalizing  $b_p(\mathbf{x}_p)$ , would be at least at a local minimum of the corresponding *Bethe* free energy [11] of the (conditional) MRF. The domain of  $\mathbf{x}_p$  is in practice discrete since the probability distribution is concentrated only at the candidate patches given by  $\mathcal{K}_{l_p}$ . Thus, every full iteration has complexity  $\mathcal{O}(K^2)$ . Using this approximation, the E-step is then given by:

$$P(t_p, l_p | \mathcal{Y}) \propto \sum_{\mathbf{x}_p} p(\mathbf{x}_p | \mathcal{Y}) P(t_p | \mathbf{x}_p) P(l_p) p(\mathbf{y}_p | l_p, t_p). \quad (4)$$

Even though loopy belief propagation is not exact (clearly, since this problem cannot be solved exactly) and not guaranteed to converge, some recent work supports this approximation [12, 13]. Other approximations include [14, 15, 1].

### 3.2. Learning the Filter Parameters

The M-step consist of optimizing the expected value of the model joint distribution under  $P(t_p, l_p | \mathbf{y}_p)$  with respect to

the model parameters  $\Lambda_l$  and  $\Psi_p$ . This can be done by computing first derivatives, as in a MAP estimate setting. For linear filters  $\Lambda_k$ , we can obtain closed form solutions for their update, likewise for the patch variances  $\Psi_p$ . The update is similar to the weighted linear regression solution. For non-linear filters, we would need to use non-linear optimization methods.

#### 4. EXPERIMENTS

We illustrate the inference and learning capabilities of our algorithm on a few image reconstruction tasks (due to printing resolution limitations, images are better viewed directly on the computer screen). We use PCA to encode the patches to account for numerical stability, speed, and storage constraints. In Fig. 2, a brick wall image was degraded by different processes (blurring, pixelation, non-linear shift, correlated and uncorrelated noise) using GIMP (a known image manipulation tool). We use a small, similar brick image as our source  $\mu$  to obtain the restored image shown. This task is perhaps simplified by the repetitive brick pattern; a few brick examples are enough for a good image restoration. Thus, in Fig. 3 we show the results of a more challenging experiment. The input image  $\mathcal{Y}$  is a painting corrupted by different types of noise in some regions and severely blurred/pixelated in other regions. We took advantage of the on-line availability of other paintings by the same painter (displaying a similar style); by using these source images, we were able to infer an image that is very close to ground truth. The different estimated filters learned to operate in localized areas. The application of the learned filters are shown in the second row of Fig. 3. Our last experiment is in image superresolution. Given a downsampled image, the objective is to obtain a high resolution equivalent of it by accounting for the missing high frequency details. Our model can be readily employed to provide a solution for superresolution, without explicitly specifying the special properties of this particular task. In Fig. 4 given a severely degraded image  $\mathcal{Y}$ , we used several high resolution images of various female models as sources to improve the high frequency details in  $\mathcal{Y}$ . The estimated filters were blurring-type filters. Other useful applications include artistic tasks such as image style translation and texture transfer [16].

#### 5. CONCLUSIONS

We presented a general method for image processing that uses example statistics from source images to reconstruct or transform an input (degraded) image and that at the same time estimates the single or multiple degradation processes undergone by this image. This approach has many uses in image restoration, reconstruction, and coding, and also in artistic applications such as non-photorealistic rendering, texture generation, and style transfer. Our method does not



**Fig. 4.** Input image (left) and inferred image (MAP)(right).

require to manually or explicitly specify the type of task required, but it allows to employ examples for this purpose.

#### 6. REFERENCES

- [1] D. Geman and S. Geman, "Stochastic relaxation, gibbs distribution and bayesian restoration of images," *IEEE Trans. PAMI*, vol. 6, no. 6, pp. 721–741, 1984.
- [2] Y. Weiss, "Interpreting images by propagating bayesian beliefs," in *NIPS*, 1997, vol. 9, p. 908.
- [3] B. Freeman, E. Pasztor, and O. Carmichael, "Learning low-level vision," *IJCV*, 2000.
- [4] A. Hertzmann, C. Jacobs, N. Oliver, B. Curless, and D. Salesin, "Image analogies," in *SIGGRAPH 2001*.
- [5] S. Zhu and D. Mumford, "Prior learning and gibbs reaction-diffusion," *IEEE Trans. PAMI*, 1997.
- [6] R. Schultz and R. Stevenson, "A bayesian approach to image expansion for improved definition," *IEEE Transactions on Image Processing*, vol. 3, no. 3, pp. 233–242, May 1994.
- [7] S. L. Lauritzen and N. Wermuth, "Graphical models for associations between variables, some of which are qualitative and some quantitative," *Annals of Statistics*, vol. 17, 1989.
- [8] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*, Morgan-Kaufman, 1988.
- [9] I. Csiszar and G. Tusnady, "Information geometry and alternating minimization procedures," *Statistics and Decisions*, vol. 1, pp. 205–237, 1984.
- [10] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood estimation from incomplete data," *Journal of the Royal Statistical Society (B)*, vol. 39(1), 1977.
- [11] J. Yedidia, "An idiosyncratic journey beyond mean field theory," *Advanced Mean Field Methods*, pp. 21–36, 2001.
- [12] F. Kschischang and B. Frey, "Iterative decoding of compound codes by probability propagation in graphical models," *J. Sel. Areas in Communications*, vol. 16, 1998.
- [13] R. McEliece, D. MacKay, and J. Cheng, "Turbo decoding as an instance of pearl's belief propagation algorithm," *J. Sel. Areas in Communications*, vol. 16, 1998.
- [14] M. Wainwright, T. Jaakkola, and A. Willsky, "A new class of upper bounds on the log partition function," in *Uncertainty in Artificial Intelligence*, 2002.
- [15] M. Jordan, Z. Ghahramani, T. Jaakkola, and L. Saul, "An introduction to variational methods for graphical models," *Learning in Graphical Models*, M. Jordan (editor), 1998.
- [16] R. Rosales, K. Achan, and B. Frey, "Unsupervised image translation," in *Proceedings ICCV*, 2003.