2012

Primal Sketch & Video Primal Sketch – Methods to Parse Images & Videos

Yuanlu Xu, SYSU, China
merayxu@gmail.com
2012.4.7

Episode 1

Backgrounds, Intuitions, and Frameworks

Background of image modeling

texton (token) vs. texture (Julesz, Marr)

Figure 1: Natural image with interweaving textures and structures.

Julesz:
Texton -> bars, edges, terminators
Texture -> sharing common statistics on certain features

Marr:
model parsimonious, enough to reconstruct

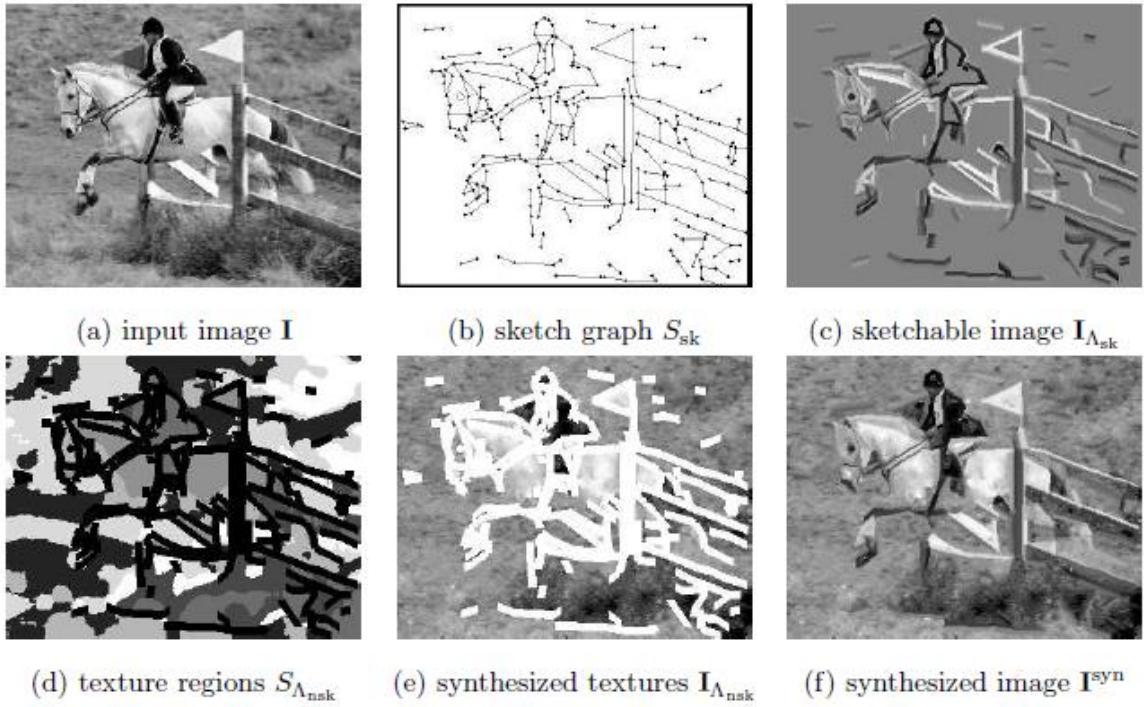
Background of image modeling

Figure 1: Natural image with interweaving textures and structures.

Texton modeling -- over-complete dictionary theory: wavelets, Fourier, ridgelets, image pyramids, and sparse coding.

Texture modeling -- Markov random field (MRF): FRAME.

Intuition of Primal Sketch



(a) input image $\mathbf{I}$ (b) sketch graph S_{sk} (c) sketchable image $\mathbf{I}_{\Lambda_{\text{sk}}}$

(d) texture regions $S_{\Lambda_{\text{nsk}}}$ (e) synthesized textures $\mathbf{I}_{\Lambda_{\text{nsk}}}$ (f) synthesized image $\mathbf{I}^{\text{syn}}$

Primal Sketch:

Sketchable vs. non-sketchable

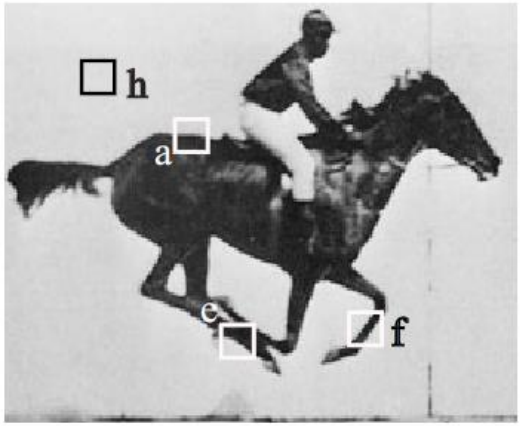
Sketchable: primitive dictionary
Non-sketchable: simplified FRAME model

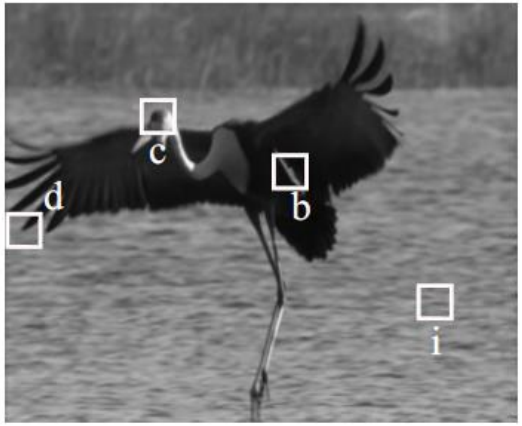
Background of video modeling

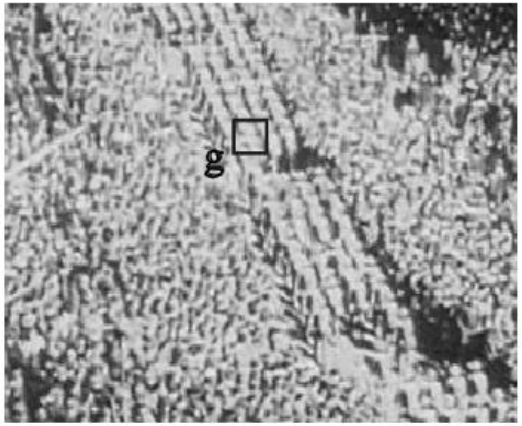
4 types of regions

Trackable motion: kernel tracking, contour tracking, key-point tracking

Intrackable motion (textured motion): dynamic texture (DT), STAR, ARMA, LDS







	Sketchable	Non-sketchable
Trackable	(a) Moving Edge (b) Moving Bar (c) Moving Blob (d) Moving Corner	(g) Moving Kernel
Intrackable	(e) High-speed Moving Edge (f) High-speed Moving Bar	(h) Flat Area (i) Textured Motion

Figure 1. The four types of local video phenomenon characterized by two criteria, sketchability and trackability .

Background of video modeling

Intrackability: Characterizing Video Statistics and Pursuing Video Representations

Haifeng Gong, Song-Chun Zhu

Definition 1 (video intrackability) Intrackability of a video sequence $\mathbf{I}_\Lambda[\tau]$ for a representation W is defined by,

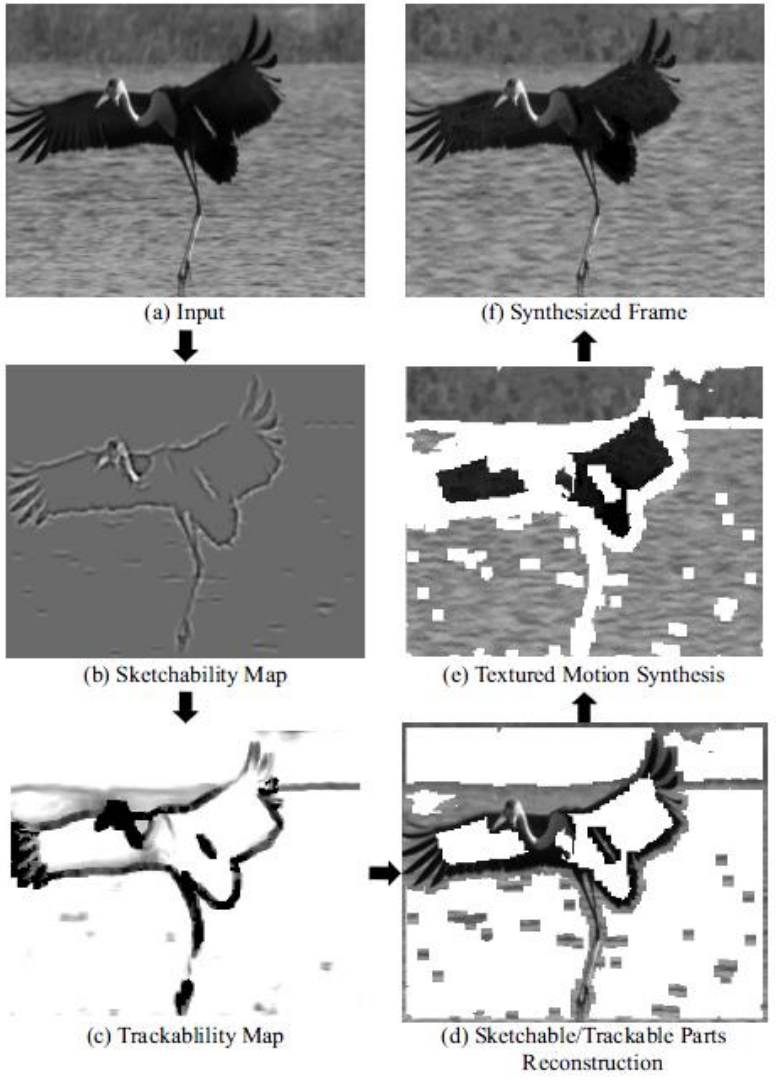
$$\mathcal{H}\{W|\mathbf{I}[\tau]\} = -\sum_{W} p(W|\mathbf{I}[\tau]) \log p(W|\mathbf{I}[\tau]). \qquad (2)$$

Intuition of Video Primal Sketch

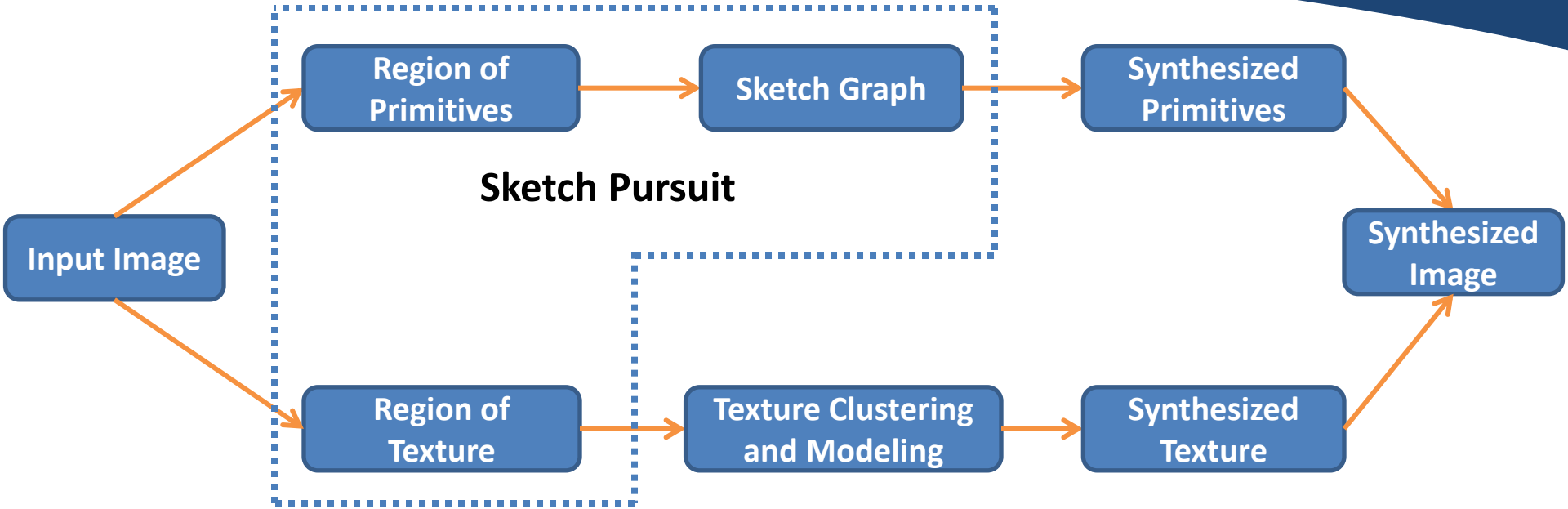
Category 4 regions into two classes: implicit regions, explicit region.

Explicit region:
sketchable and trackable, sketchable and non-trackable, non-sketchable and trackable
Modeling with sparse coding

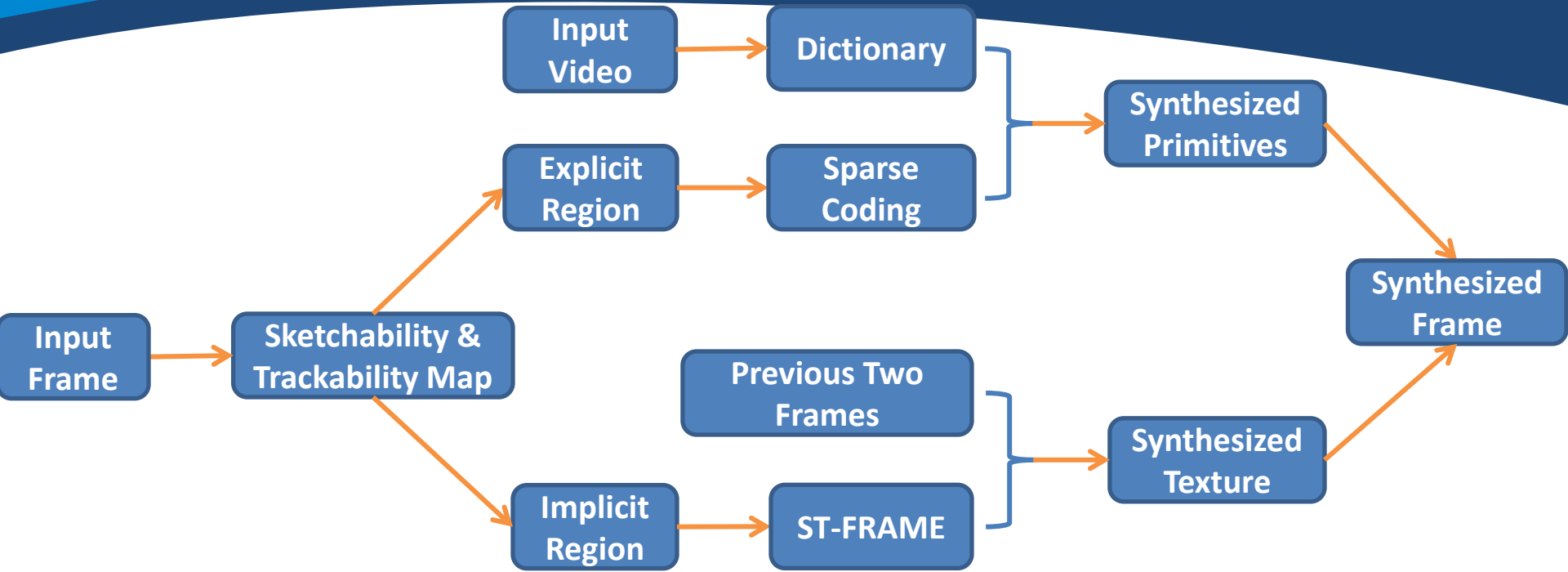
Implicit region:
Non-sketchable and non-trackable
Modeling with ST-FRAME



The Framework of Primal Sketch



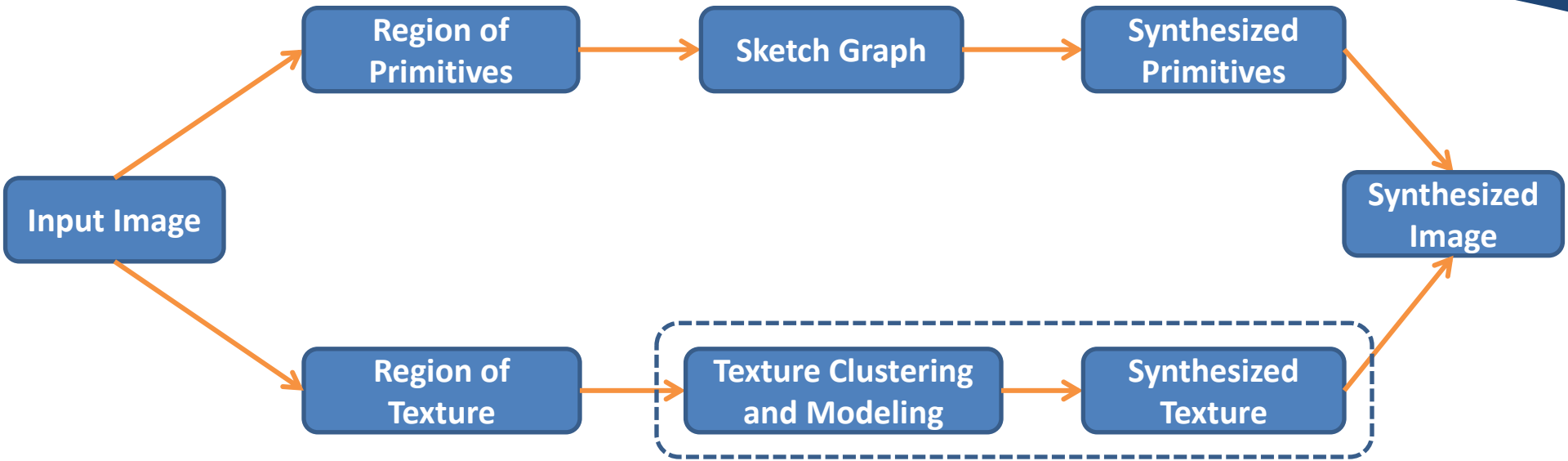
The Framework of Video Primal Sketch



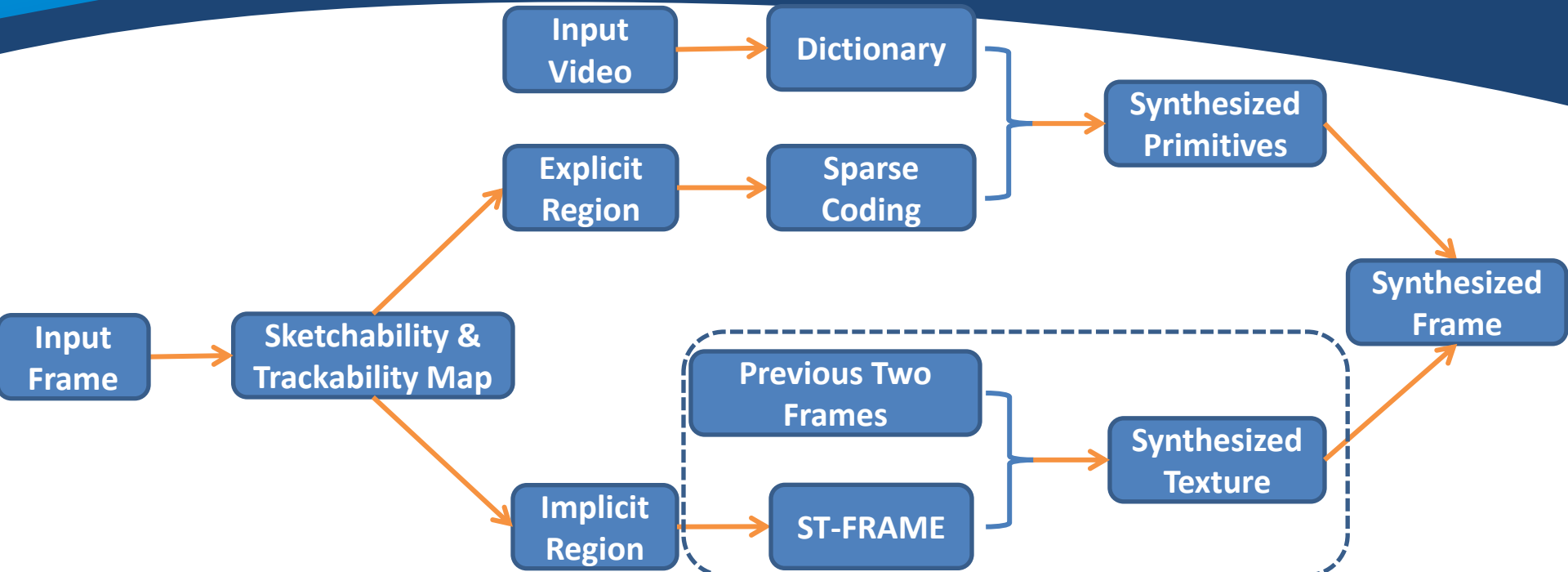
Episode 2

Texture Modeling

The Framework of Primal Sketch



The Review of Video Primal Sketch

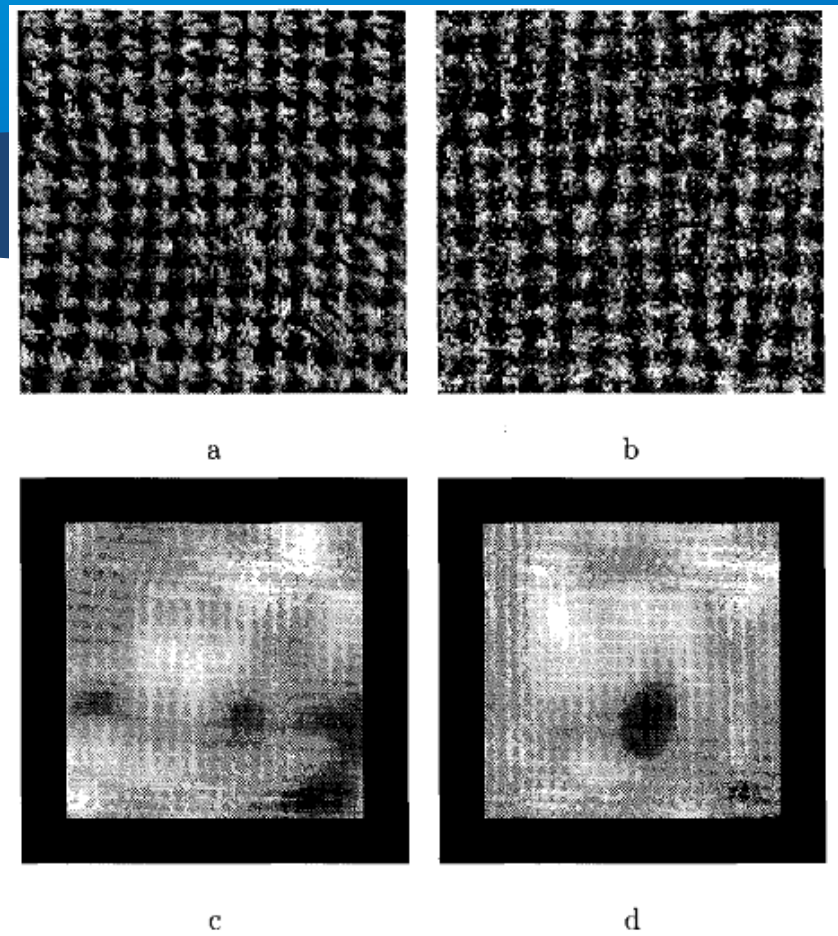


FRAME - Overview

Filters, Random Fields and Maximum Entropy (FRAME): Towards a Unified Theory for Texture Modeling

Songchun Zhu, Yingnian Wu, David Mumford IJCV 1998

Texture: a set of images sharing common statistics on certain features.



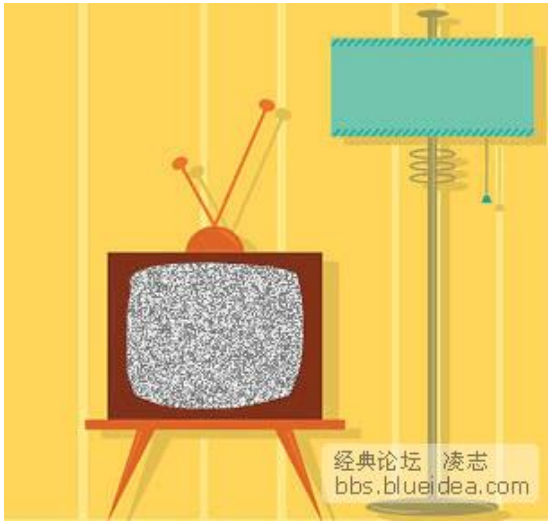
FRAME - Minimax Entropy Principle

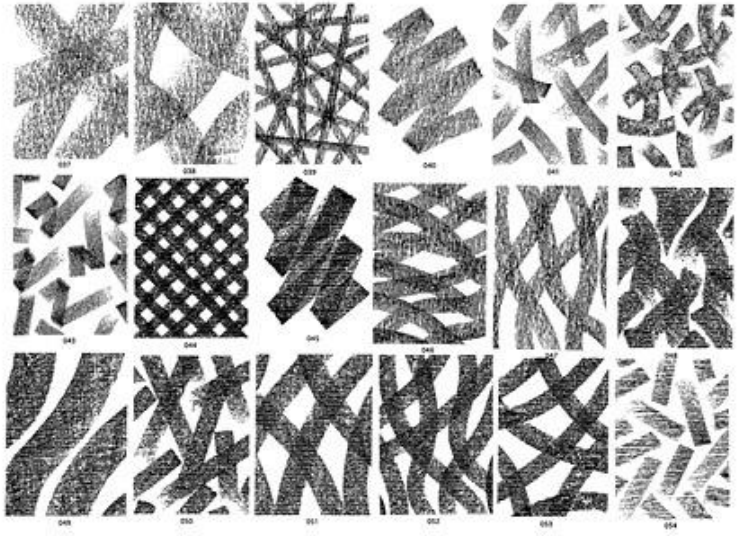
f(I): underlying probability of a texture,
p(I): estimate probability distribution of f(I) from an textured image.

a. Entropy $\text{entropy}(p(I)) = -\int p(I) \log p(I)\, dI$ stands for the expected coding length. On the other hand, entropy is the negative Kullback-Leibler distance, up to a constant, between $p(I)$ and the uniform distribution, the latter stands for noise images. To minimize the entropy, $p(I)$ should be made as "orderly" (or far away from the uniform distribution) as possible.

FRAME - Minimax Entropy Principle

b. To constrain the complexity, we choose an optimal set of features, while it has the minimum entropy. Denoted the feature set S_n, the set of all possible probability distributions $p(I)$ that satisfy the constraints in S_n as Ω_n.





FRAME - Minimax Entropy Principle

c. The maximum entropy principle suggests that the probability distribution in Ω_n with maximum entropy is the best estimate of $p(I)$.

FRAME - Minimax Entropy Principle

10 Principle of Maximum Entropy

In Bayesian probability theory, the principle of maximum entropy is an axiom. It states that, subject to precisely stated prior data, which must be a proposition that expresses testable information, the probability distribution which best represents the current state of knowledge is the one with largest information theoretical entropy.

Let some precisely stated prior data or testable information about a probability distribution function be given. Consider the set of all trial probability distributions that encode the prior data. Of those, the one that maximizes the information entropy is the proper probability distribution under the given prior data.

FRAME - Minimax Entropy Principle

d. The minimax principle means that $p^*(I)$ should satisfy the constraints and as "orderly" as possible along some dimensions Ω_n, and should also be as random as possible in other unconstrained dimensions.

FRAME - Minimax Entropy Principle

e. The problem is reformed as follows

$$\text{maximize} \; -\int p(I) \log p(I) dI,$$

$$\text{subject to} \int \phi_i(I) f(I) dI = \mu_i, \quad i = 1, \ldots, n.$$

FRAME - Minimax Entropy Principle

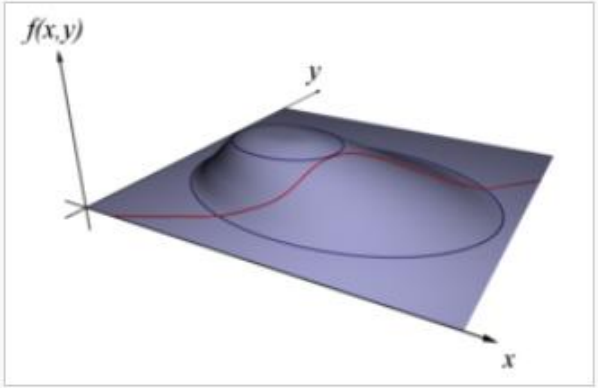


Figure 1: Find *x* and *y* to maximize $f(x, y)$ subject to a constraint (shown in red) $g(x, y) = c$

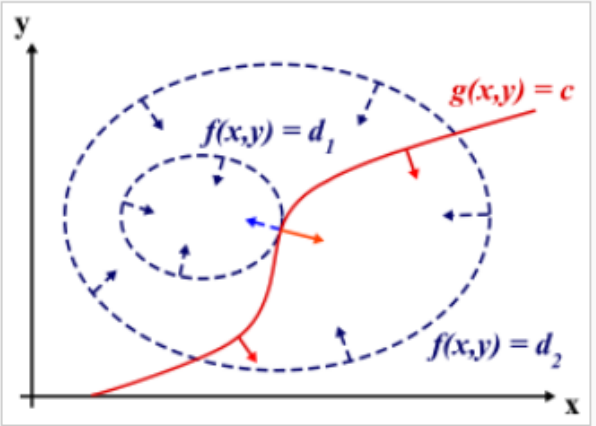


Figure 2: Contour map of Figure 1. The red line shows the constraint $g(x, y) = c$. The blue lines are contours of $f(x, y)$. The point where the red line tangentially touches a blue contour is our solution.

FRAME - Minimax Entropy Principle

A point on f is a constrained stationary point if and only if the direction that changes f violates at least one of the constraints.

$$\left.\begin{array}{l} \nabla f \cdot v = 0 \\ \nabla g \cdot v = 0 \end{array}\right\} \quad \nabla f(p) = \lambda \nabla g(p) \quad \Rightarrow \quad \nabla f(p) - \lambda \nabla g(p) = 0$$

FRAME - Minimax Entropy Principle

To satisfy multiple constraints we can state that at the stationary points, the direction that changes *f* is in the "violation space" created by the constraints acting jointly.

That is, a stationary point satisfies:

$$\begin{aligned} g_1(p) &= 0 \\ g_2(p) &= 0 \\ &\vdots \\ g_M(p) &= 0 \end{aligned}$$

these mean the point satisfies all constraints

$$\nabla f(p) - \sum_{k=1}^{M} \lambda_k \nabla g_k(p) = 0$$

this means the point is a stationary point

FRAME - Minimax Entropy Principle

e. The problem is reformed as follows

$$\text{maximize} \; -\int p(I) \log p(I) dI,$$

$$\text{subject to} \int \phi_i(I) f(I) dI = \mu_i, \quad i = 1, \ldots, n.$$

FRAME - Minimax Entropy Principle

According to Lagrange multipliers, the identification of stationary points is

$$\int \phi_i(I) f(I) dI - \mu_i = 0, \quad i = 1, \ldots, n$$

$$\nabla(-\int p(I) \log p(I) dI) - \sum_{i=1}^{n} \lambda_i \nabla(\int \phi_i(I) f(I) dI - \mu_i) = 0$$

The solution $(\lambda_1, \ldots, \lambda_n)$ is deduced as follows

$$\nabla(-\int p(I) \log p(I) dI) - \sum_{i=1}^{n} \lambda_i \nabla(\int \phi_i(I) f(I) dI - \mu_i) = 0$$

$$\Leftrightarrow \frac{\partial(-\int p(I) \log p(I) dI)}{\partial I} - \sum_{i=1}^{n} \lambda_i \frac{\partial(\int \phi_i(I) f(I) dI - \mu_i)}{\partial I} = 0$$

$$\Leftrightarrow -p(I) \log p(I) - \sum_{i=1}^{n} \lambda_i \phi_i(I) f(I) = 0$$

$$\Leftrightarrow \log p(I) = -\sum_{i=1}^{n} \lambda_i \phi_i(I)$$

$$\Leftrightarrow p(I) = \frac{1}{Z} e^{-\sum_{i=1}^{n} \lambda_i \phi_i(I)},$$

FRAME - Minimax Entropy Principle

where $Z = \int e^{-\sum_{i=1}^{n} \lambda_i \phi_i(I)} dI$ is the partition function.

Function Z has the following nice properties:

$$i) \quad \frac{\partial \log Z}{\partial \lambda_i} = \frac{1}{Z}\frac{\partial Z}{\partial \lambda_i} = -E_p[\phi_i(I)] = -\mu_i,$$

$$ii) \quad \frac{\partial^2 \log Z}{\partial \lambda_i \partial \lambda_j} = E_p[(\phi_i(I) - \mu_i)(\phi_j(I) - \mu_j)].$$

Property 2 tells us the Hessian matrix of function *log Z* is the covariance matrix of *log Z* and is positive definite. Therefore, *Z* is log concave. It is easy to prove *log p(x)* is convex, either. Given a set of consistent constraints, the solution for $(\lambda_1, \ldots, \lambda_n)$ is unique.

FRAME - Minimax Entropy Principle

Considering a closed form solution is not available in general, we seek numerical solutions by solving the following equations iteratively.

$$\frac{d\lambda_i}{dt} = E_{p(I;\Lambda)}[\phi_i(I)] - \mu_i,\ i = 1, \ldots, n.$$

Gradient Descent

FRAME - Minimax Entropy Principle

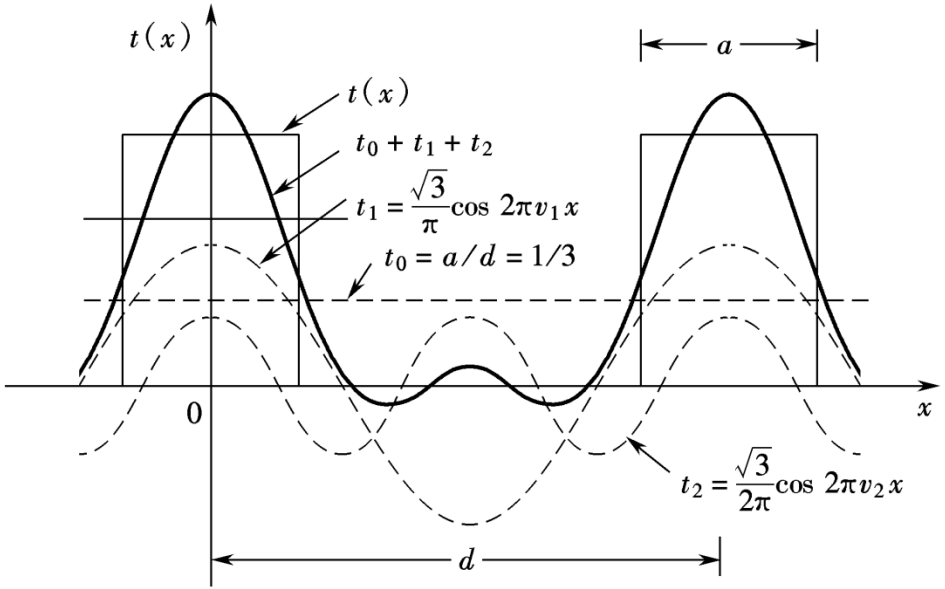
f. In summary, given the model complexity n, an optimal probability model $p(I)$ or equivalently an optimal probability model $p(I)$ should be derived from the following criterion.

$$p^*(I) = \arg\min_{S_n \in S} \{ \arg\max_{p \in \Omega_n} \text{entropy}(p(I)) \} \tag{4}$$

FRAME – Deriving the FRAME Model

a. To reduce the dimensionality of the distribution $f(I)$. $f(I)$ is transformed into the linear combination of one dimensional marginal distributions. The author proves that if the marginal distributions of $F^{(\xi)} * I$ for all ξ are matched, the underlying distribution $f(I)$ can be eventually matched. Considering the complexity of the model, a fixed number of filters are employed to represent $f(I)$.

Fourier transformation



FRAME – Deriving the FRAME Model

b. Three assumptions are proposed to further constrain the complexity of FRAME model.

1. Texture discrimination can be captured by the locally supported filters $F^{(\alpha)}$.
2. The texture is homogenous such that $f(I)$ is translation invariant with respect to the pixel location $\vec{v}$.
3. For any probability distribution $p(I)$, if $p(I)$ has the same marginal distribution $f^{\alpha}(z)$ as $f(I)$, for all $\alpha = 1, 2, \ldots, K$, then $p(I)$ is considered to be perceptually a good enough approximation to $f(I)$.

FRAME – Deriving the FRAME Model

Given an image $\mathbf{I}$ and a filter $F^{(\alpha)}$ with $\alpha = 1, 2, \ldots, K$ being an index of filter, we let $\mathbf{I}^{(\alpha)}(\vec{v}) = F^{(\alpha)} * \mathbf{I}(\vec{v})$ be the filter response at location $\vec{v}$, and $\mathbf{I}^{(\alpha)}$ the filtered image. The marginal empirical distribution (histogram) of $\mathbf{I}^{(\alpha)}$ is

$$H^{(\alpha)}(z) = \frac{1}{|\mathcal{D}|} \sum_{\vec{v} \in \mathcal{D}} \delta\big(z - \mathbf{I}^{(\alpha)}(\vec{v})\big),$$

where $\delta(\ )$ is the Dirac delta function. The marginal distribution of $f(\mathbf{I})$ with respect to $F^{(\alpha)}$ at location $\vec{v}$ is denoted by

$$f_{\vec{v}}^{(\alpha)}(z) = \int \int_{\mathbf{I}^{(\alpha)}(\vec{v})=z} f(\mathbf{I}) d\mathbf{I} = E_f\big[\delta\big(z - \mathbf{I}^{(\alpha)}(\vec{v})\big)\big].$$

The Dirac delta can be loosely thought of as a function on the real line which is zero everywhere except at the origin, where it is infinite,

$$\delta(x) = \begin{cases} +\infty, & x = 0 \\ 0, & x \neq 0 \end{cases}$$

and which is also constrained to satisfy the identity

$$\int_{-\infty}^{\infty} \delta(x)\, dx = 1$$

FRAME – Deriving the FRAME Model

c. The constraints set $\Omega = \{p(I)|E_p[\delta(I^{(\alpha)}(\vec{v}) - z)] = f^{(\alpha)}(z) \;\; \forall z \, \forall \alpha \, \forall \vec{v}\}$ defines that z takes continuous real values, hence there are infinite number of constraints and λ takes the form as a function of z. Assumed that the filter responses $I^{(\alpha)}$ are quantified into L discrete values, and the model can be represented as

$$p^*(I) = \frac{1}{Z} e^{\sum_{\vec{v}} \sum_{\alpha=1}^{K} \sum_{i=1}^{L} \lambda_i^{(\alpha)} \delta(I^{\alpha}(\vec{v}) - z_i^{(\alpha)})}, \tag{6}$$

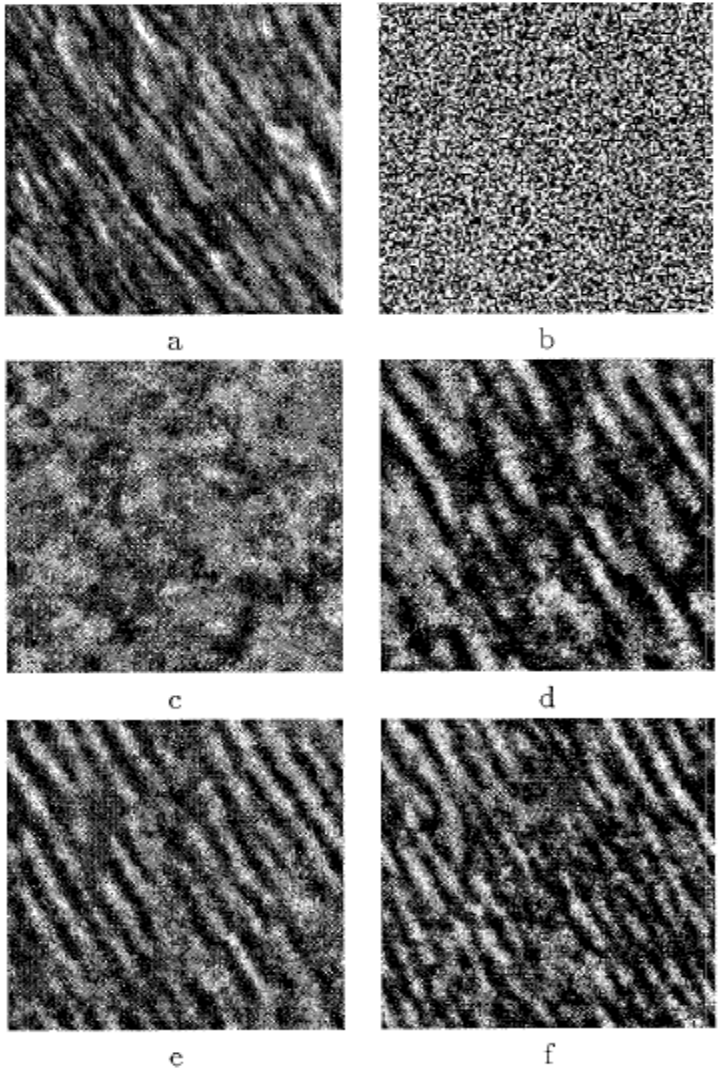
changing the order of summations, we get:

$$p^*(I) = \frac{1}{Z} e^{\sum_{\alpha=1}^{K} \sum_{i=1}^{L} \lambda_i^{(\alpha)} H_i^{(\alpha)}}, \tag{7}$$

FRAME – Deriving the FRAME Model

Plugging the above equation into the constraints of Maximum Entropy distribution, we get

$$\frac{d\lambda^{(\alpha)}}{dt} = \frac{1}{Z}\frac{\partial Z}{\partial \lambda^{(\alpha)}} - H^{obs(\alpha)} = E_p(H^{(\alpha)}) - H^{obs(\alpha)},$$



FRAME – Choice of Filters

k is the number of filters selected to model f(I) and p_k(I) the best estimate of f(I) given k filters

a. Kullback-Leibler distance is applied to measure the difference between $p_k(I)$ and $f(I)$:

$$D(f, p_k) = \int f(I) \log \frac{f(I)}{p_k(I)} dI = E_f[\log f(I)] - E_f[\log p_k(I)]. \quad (10)$$

Based on the definition of entropy, $D(f, p_k)$ can be computed by

$$D(f, p_k) = \text{entropy}(p_k(I)) - \text{entropy}(f(I)). \quad (11)$$

FRAME – Choice of Filters

b. The desired filters are chosen by a stepwise greedy strategy. At the k-th step, Suppose $S_k = \{F^{(1)}, F^{(2)}, \ldots, F^{(k)}\}$ has been selected from the filter bank B. Then at the $(k+1)$-th step, the $(k+1)$-th filter is chosen from the rest of the filter bank according to the criterion below,

$$F^{(k+1)} = \underset{F^{(\beta)} \in B/S_k}{\arg\max} \frac{1}{2} | H^{obs(\beta)} - H^{syn(\beta)} |. \tag{12}$$

FRAME – Choice of Filters

Constructing a filter bank B using five kinds of filters

1. The intensity filter $\delta(\,)$, and it captures the DC component.
2. The isotropic center-surround filters, i.e., the Laplacian of Gaussian filters.
3. The Gabor filters with both sine and cosine components.
4. The spectrum analyzers denoted by $SP(T, \theta)$, whose responses are the power of the Gabor pairs: $|(\text{Gabor} * \mathbf{I})(x, y)|^2$.
5. Some specially designed filters for one dimensional textures and the textons.

FRAME – Synthesized Texture

Gibbs sampling or a Gibbs sampler is an algorithm to generate a sequence of samples from the joint probability distribution of two or more random variables.

The purpose of such a sequence:

1. approximate the joint distribution;
2. approximate the marginal distribution of one of the variables, or some subset of the variables;
3. compute an integral (such as the expected value of one of the variables).

FRAME – Synthesized Texture

- Another MCMC Method
- Update a single parameter at a time
- Sample from conditional distribution when other parameters are fixed

FRAME – Synthesized Texture

Consider a particular choice of parameter values $\boldsymbol{\theta}^{(t)}$

Define the next set of parameter values by :

a. Selecting component to update, say i

b. Sample value for $\theta_i^{(t+1)}$ from $p(\theta_i \mid x, \theta_1, \theta_2, \ldots \theta_{i-1}, \theta_{i+1}, \ldots \theta_k)$

Increment t and repeat previous steps.

FRAME – Synthesized Texture

Consider a particular choice of parameter values $\boldsymbol{\theta}^{(t)}$

Define the next set of parameter values by :

a. Update each component, 1 .. k, in turn

b. Sample value for $\theta_1^{(t+1)}$ from $p(\theta_1 \mid x, \theta_2, \theta_3, \ldots \theta_k)$

c. Sample value for $\theta_2^{(t+1)}$ from $p(\theta_2 \mid x, \theta_1, \theta_3, \ldots \theta_k)$

...

z. Sample value for $\theta_k^{(t+1)}$ from $p(\theta_k \mid x, \theta_1, \theta_3, \ldots \theta_{k-1})$

Increment t and repeat previous steps.

FRAME – Synthesized Texture

Suppose that $(\theta_1^{(t)}, \theta_2^{(t)}, ..., \theta_k^{(t)}) \sim p(\theta_1, \theta_2, ..., \theta_k \mid x)$

Then $(\theta_1^{(t+1)}, \theta_2^{(t)}, ..., \theta_k^{(t)})$ is distributed as

$$p(\theta_1 \mid \theta_2, ..., \theta_k, x)\, p(\theta_2, ..., \theta_k \mid x) = p(\theta_1, \theta_2, ..., \theta_k \mid x)$$

In fact...

$$\boldsymbol{\theta}^{(t)} \sim p(\boldsymbol{\theta} \mid x) \Rightarrow \boldsymbol{\theta}^{(t+1)} \sim p(\boldsymbol{\theta} \mid x)$$

is not a function of θ_1 and thus is the same for all values of θ_1

Eventually, we expect the Gibbs sampler to sample parameter values from their posterior distribution

FRAME – Synthesized Texture

Algorithm 2. The Gibbs Sampler for w Sweeps

Given image $\mathbf{I}(\vec{v})$, flip_counter $\leftarrow 0$

Repeat

  Randomly pick a location $\vec{v}$ under the uniform distribution.

  For val $= 0, \ldots, G-1$ with G being the number of grey levels of $\mathbf{I}$

    Calculate $p(\mathbf{I}(\vec{v}) = \text{val} \mid \mathbf{I}(-\vec{v}))$ by $p(\mathbf{I}; \Lambda_K, S_K)$.

  Randomly flip $\mathbf{I}(\vec{v}) \leftarrow$ val under $p(\text{val} \mid \mathbf{I}(-\vec{v}))$.

  flip_counter $\leftarrow$ flip_counter $+ 1$

Until flip_counter $= w \times M \times N$.

In Algorithm 2, to compute $p(\mathbf{I}(\vec{v}) = \text{val} \mid \mathbf{I}(-\vec{v}))$, we set $\mathbf{I}(\vec{v})$ to val, due to Markov property, we only need to compute the changes of $\mathbf{I}^{(\alpha)}$ at the neighborhood of $\vec{v}$. The size of the neighborhood is determined by the size of filter $F^{(\alpha)}$. With the updated $\mathbf{I}^{(\alpha)}$, we calculate $H^{(\alpha)}$, and the probability is normalized such that $\sum_{\text{val}=0}^{G-1} p(\mathbf{I}(\vec{v}) = \text{val} \mid \mathbf{I}(-\vec{v})) = 1$.

FRAME – Detailed Framework

Algorithm 1: FRAME Model

Input: Input textured image I^{obs} (image size $M \times N$, gray level G), bank of filters B

Output: Probability distribution of textured image $p(I)$, synthesized textured image I^{syn}

Initialize $k = 0$, $S_0 = \emptyset$, $p_0(I) \leftarrow$ uniform distribution $I^{syn} \leftarrow$ uniform white noise image;

for $\alpha = 1, \ldots, |B|$ **do**

  Compute $I^{obs(\alpha)}$ by applying F^{α} to I^{obs} ;

  Compute Histogram $H^{obs(\alpha)}$ of $I^{obs(\alpha)}$;

end

repeat

  foreach $F^{(\beta)} \in B/S_k$ **do**

    Compute $I^{syn(\beta)}$ by applying $F^{(\beta)}$ to I^{syn} ;

    Compute histogram $H^{syn(\beta)}$ of $I^{syn(\beta)}$;

    $d(\beta) = \frac{1}{2} |H^{obs(\beta)} - H^{syn(\beta)}|$;

  end

  Choose the filter $F^{(k+1)}$ according to $d(k+1) = \max\{d(\beta),\ F^{(\beta)} \in B/S_k\}$;

  $S_{k+1} \leftarrow F^{(k+1)} \bigcup S_k$, $k \leftarrow k+1$;

  Initialize $\lambda^{(\alpha)} \leftarrow 0$, $\alpha = 1, \ldots, k$;

FRAME – Detailed Framework

repeat
  repeat
    Calculate $H^{syn(\alpha)}, \alpha = 1, 2, \ldots, k$ from I^{syn};
    Update $\lambda^{\alpha}, \alpha = 1, 2, \ldots, k$ and $p(I; \Lambda_k, S_k)$ is updated;
    Initialize flip counter $c \leftarrow 0$, sweep times ω ;
    repeat
      Randomly pick a location $\vec{v}$ in I^{syn} ;
      for $val = 0, 1, \ldots, G-1$ **do**
        Calculate $p(\, I(\vec{v}) = val | I(-\vec{v})\,)$ by $p(I; \Lambda_k, S_k)$;
      end
      Randomly flip $I(\vec{v}) \leftarrow val$ under $p(\, I(\vec{v}) = val | I(-\vec{v})\,)$;
      $c \leftarrow c + 1$;
    until $c = \omega \times M \times N$;
  until $\frac{1}{2} |H^{obs(\alpha)} - H^{syn(\alpha)}| < \epsilon$ *for* $\alpha = 1, 2, \ldots, k$;
until $d(k) < \epsilon$;

Simplified Version in Primal Sketch

To segment the whole texture region into small ones, the clustering process is maximizing a posterior, with the assumption that each sub-region obeying a multivariate Gaussian distribution:

$$
\begin{aligned}
S^*_{\text{nsk}} &= \arg\max p(S_{\text{nsk}}|I_{\Lambda_{\text{nsk}}}) \\
&= \arg\max p(I_{\Lambda_{\text{nsk}}}|S_{\text{nsk}})p(S_{\text{nsk}}) \\
&= \arg\min \sum_{m=1}^{M} E(I_{\Lambda_{\text{nsk},m}}|S_{\text{nsk},m}) + E(S_{\text{nsk}}), \\
E(I_{\Lambda_{\text{nsk},m}}|S_{\text{nsk},m}) &= -\frac{1}{2}\log|\Sigma_m| - \frac{1}{2}\sum_{(u,v)\in\Lambda_{\text{nsk},m}} (h(u,v)-h_m)^T\Sigma_m^{-1}(h(u,v)-h_m),
\end{aligned}
$$

where $E(S_{\text{nsk}})$ follows the Potts model

Simplified Version in Primal Sketch

$$E(S_{\text{nsk}}) = -\sum_{p_1 \sim p_2, p_1 \in \Lambda_{\text{nsk},i}, p_2 \in \Lambda_{\text{nsk},j}} \lambda_{\text{nsk}} \delta(i,j), \tag{17}$$

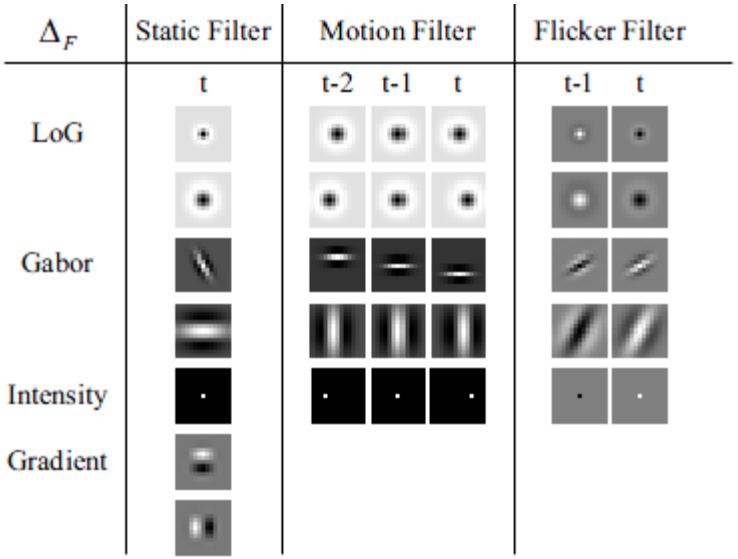
where $p_1 \sim p_2$ means that p_1 and p_2 are two pixels that are neighbors of each other. $\lambda_{\text{nsk}} (\geq 0)$ is the parameter for the Potts model which favorites identical labelling for neighboring pixels. $\delta(i,j) = 0$, if $i = j$. Otherwise $\delta(i,j) = 1$.

Simplified Version in Primal Sketch

$$
\begin{aligned}
S^*_{\text{nsk}} &= \arg\max p(S_{\text{nsk}}|I_{\Lambda_{\text{nsk}}}) \\
&= \arg\max p(I_{\Lambda_{\text{nsk}}}|S_{\text{nsk}})p(S_{\text{nsk}}) \\
&= \arg\min \sum_{m=1}^{M} E(I_{\Lambda_{\text{nsk},m}}|S_{\text{nsk},m}) + E(S_{\text{nsk}}), \\
E(I_{\Lambda_{\text{nsk},m}}|S_{\text{nsk},m}) &= -\frac{1}{2}\log|\Sigma_m| - \frac{1}{2}\sum_{(u,v)\in\Lambda_{\text{nsk},m}} (h(u,v)-h_m)^T\Sigma_m^{-1}(h(u,v)-h_m),
\end{aligned}
$$

$$
E(S_{\text{nsk}}) = -\sum_{p_1\sim p_2, p_1\in\Lambda_{\text{nsk},i}, p_2\in\Lambda_{\text{nsk},j}} \lambda_{\text{nsk}}\delta(i,j),
$$

Adapted Version in Video Primal Sketch (ST-FRAME)



Static filters. Laplacian of Gaussian (LoG), Gabor, gradient, or intensity filter on a single frame. They capture statistics of spatial features.

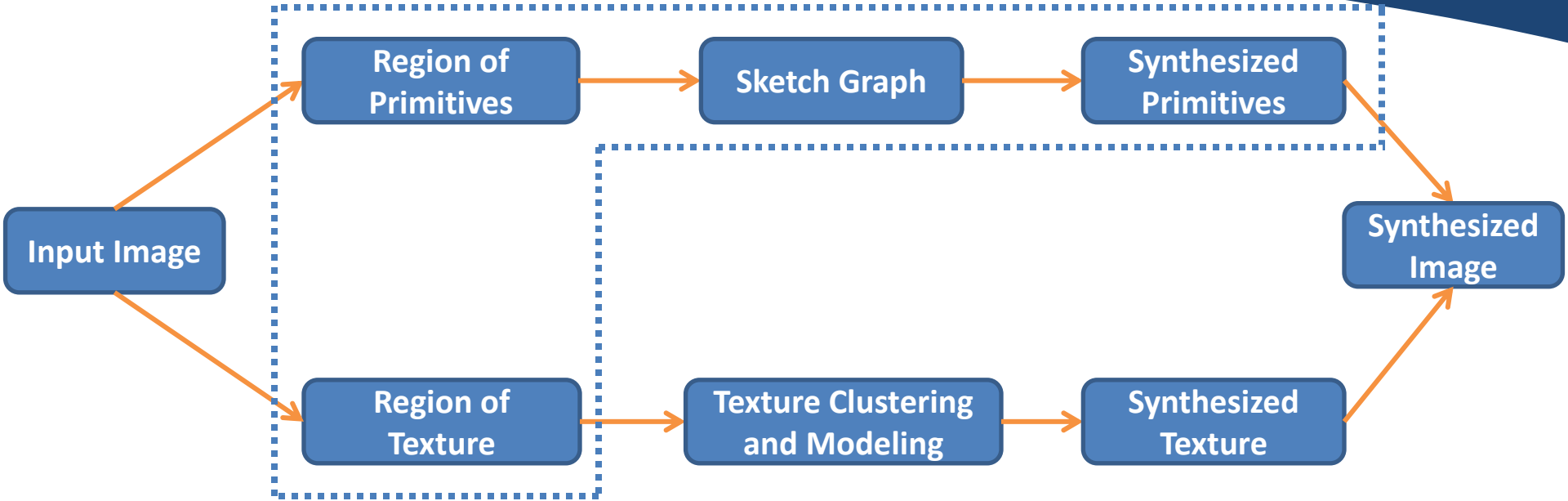
Motion filters. Moving LoG, Gabor or intensity filters in different velocities and orientations over three frames. Specifically, Gabor motion filters move perpendicularly to their orientations.

Flicker filters. One static filter with opposite signs at two frames. It contrasts the static filter responses between two consequent frames and detect the change of dynamics.

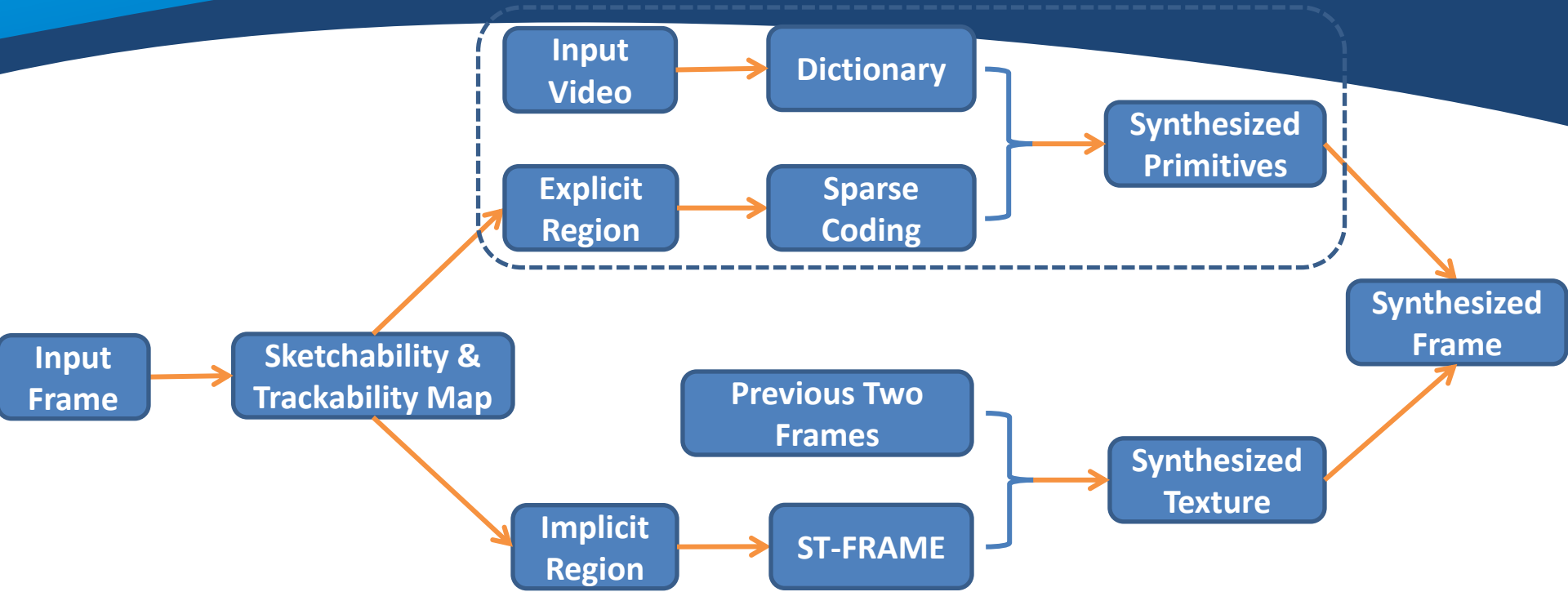
Episode 3

Texton Modeling

The Framework of Primal Sketch



The Review of Video Primal Sketch



Sparse Coding

The image coding theory assumes that I is the weighted sum of a number of *image bases* B_i indexed by *i* for its position, scale, orientation etc. Thus one obtains a "generative model",

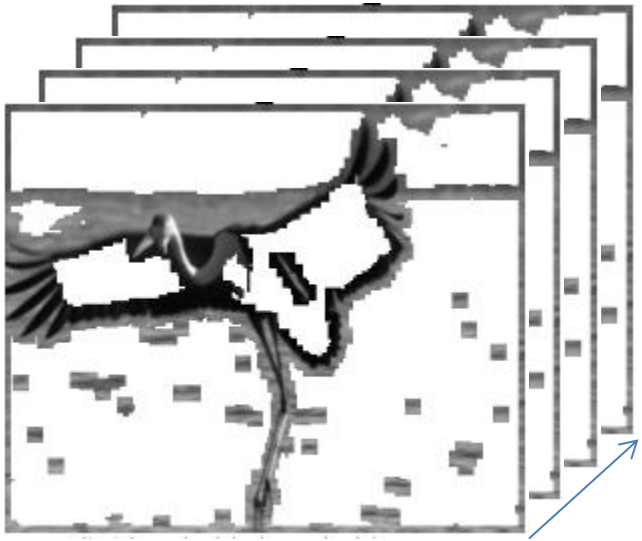
$$\mathbf{I} = \sum_{k=1}^{K} c_k B_k + \epsilon, \quad B_k \in \Delta_B,$$

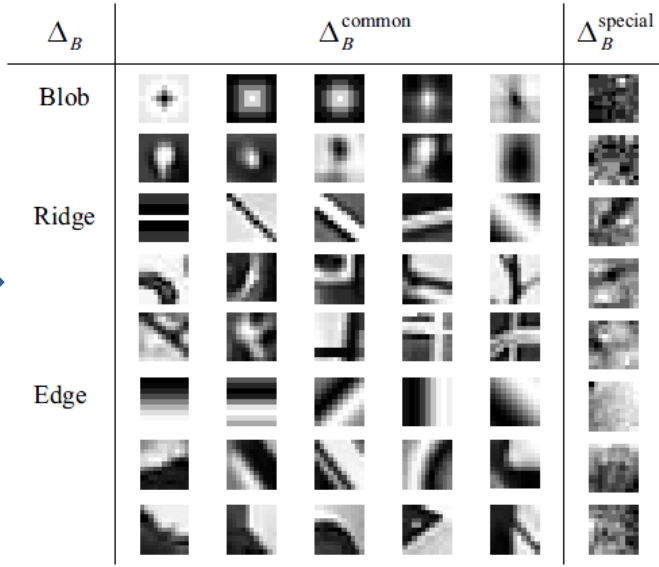
where B_k are selected from a dictionary Δ_B, c_k are the coefficients, and ϵ is the residual error modeled by Gaussian white noise.

Sparse Coding

Sparse Coding:

Definition: modeling data vectors as sparse linear combinations of basis elements.





Sparse Coding

Online Dictionary Learning for Sparse Coding
ICML 2009
Julien Mairal
Francis Bach
Jean Ponce
Guillermo Sapiro

Characteristic: **Online Dictionary Learning**
(Incremental Learning)

Sparse Coding

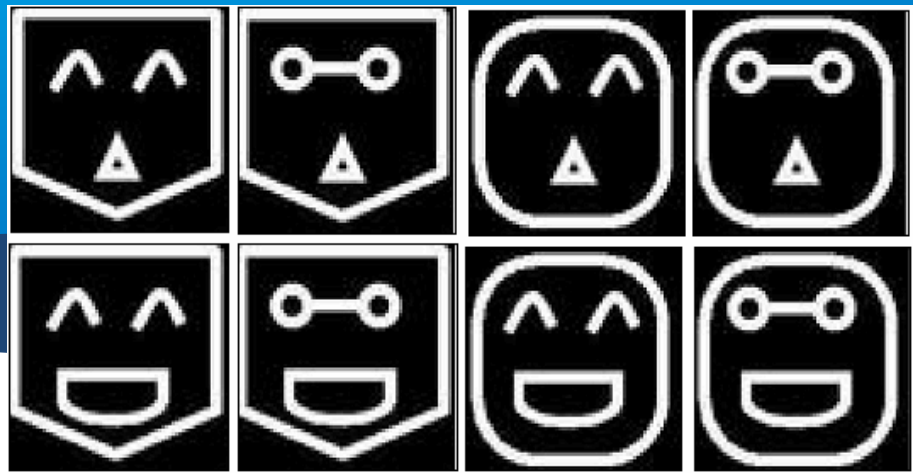
Classical Dictionary Learning:

Given a finite training set of signals $X = \{x_1, x_2, x_n\} \in \boldsymbol{R}^{m*n}$, optimize the empirical cost function:

$$f_n(\mathbf{D}) \triangleq \frac{1}{n}\sum_{i=1}^{n} l(\mathbf{x}_i, \mathbf{D}),$$

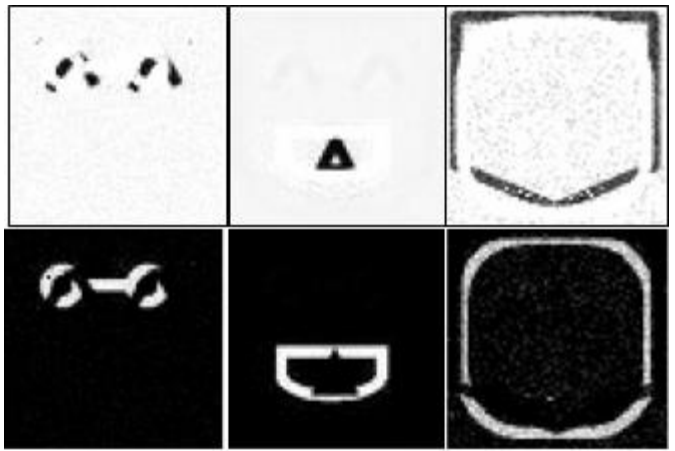
where $\boldsymbol{D} \in \boldsymbol{R}^{m*k}$ is the dictionary, each column representing a basis vector, and l is a loss function measuring the reconstruction residual.

Sparse Coding



Intuitive Explanation of Sparse Coding:

Given *n* samples with dimension of each sample *m*, usually *n* >> *m*, constructing an over-complete dictionary ***D*** with *k* bases, *k* >= *m*, each sample only uses a few bases in ***D***.



Sparse Coding

Key:

$$\text{minimize} \quad l(\mathbf{x}, \mathbf{D}) \triangleq \min_{\boldsymbol{\alpha} \in \mathbb{R}^k} \frac{1}{2} ||\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}||_2^2 + \lambda ||\boldsymbol{\alpha}||$$

where λ is a regularization parameter.

L1 – Norm Penalty

L0 – Norm Penalty : Aharon et al. (2006)

Sparse Coding

Problems of using L1 – norm penalty:

L1 – norm is not equivalent to sparsity.

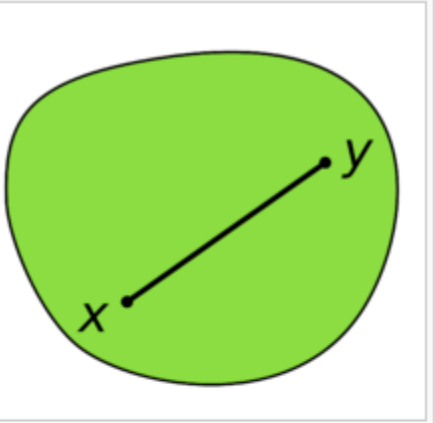
[2]The ℓ_p norm of a vector $\mathbf{x}$ in $\mathbb{R}^m$ is defined, for $p \geq 1$, by $||\mathbf{x}||_p \triangleq (\sum_{i=1}^{m} |\mathbf{x}[i]|^p)^{1/p}$. Following tradition, we denote by $||\mathbf{x}||_0$ the number of nonzero elements of the vector $\mathbf{x}$. This "ℓ_0" sparsity measure is not a true norm.

Sparse Coding

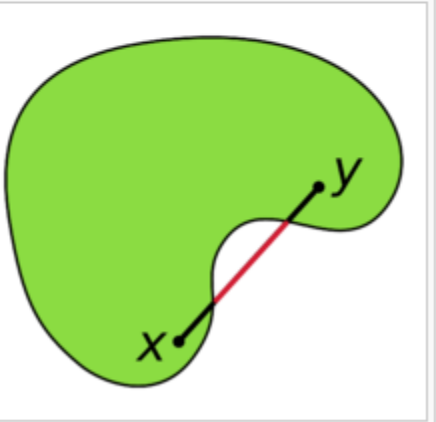
$$l(\mathbf{x}, \mathbf{D}) \triangleq \min_{\boldsymbol{\alpha} \in \mathbb{R}^k} \frac{1}{2}||\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}||_2^2 + \lambda||\boldsymbol{\alpha}||$$

To prevent D from being arbitrarily large (which would lead to arbitrarily small values of $\boldsymbol{\alpha}$), it is common to constrain its columns $(\mathbf{d}_j)_{j=1}^k$ to have an ℓ_2 norm less than or equal to one. We will call $\mathcal{C}$ the convex set of matrices verifying this constraint:

$$\mathcal{C} \triangleq \{\mathbf{D} \in \mathbb{R}^{m \times k} \text{ s.t. } \forall j = 1, \ldots, k, \ \mathbf{d}_j^T \mathbf{d}_j \leq 1\}. \quad (3)$$



A convex set.



A non-convex set, with a line-segment outside the set.

Sparse Coding

Note that the problem of minimizing the empirical cost $f_n(\mathbf{D})$ is not convex with respect to $\mathbf{D}$. It can be rewritten as a joint optimization problem with respect to the dictionary $\mathbf{D}$ and the coefficients $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_n]$ of the sparse decomposition, which is not jointly convex, but convex with respect to each of the two variables $\mathbf{D}$ and $\boldsymbol{\alpha}$ when the other one is fixed:

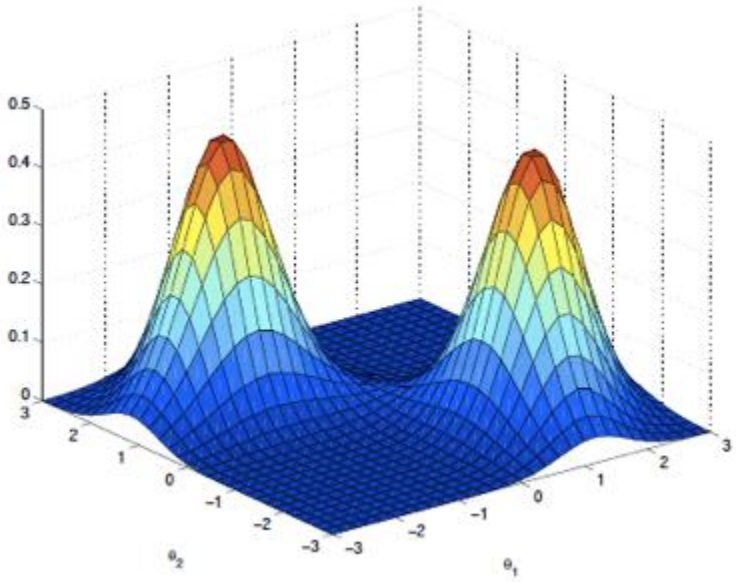
$$\min_{\mathbf{D}\in\mathcal{C},\boldsymbol{\alpha}\in\mathbb{R}^{k\times n}} \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{2}||\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i||_2^2 + \lambda||\boldsymbol{\alpha}_i||_1\right). \quad (4)$$

Sparse Coding

$$\min_{\mathbf{D}\in\mathcal{C},\boldsymbol{\alpha}\in\mathbb{R}^{k\times n}} \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{2}||\mathbf{x}_i-\mathbf{D}\boldsymbol{\alpha}_i||_2^2+\lambda||\boldsymbol{\alpha}_i||_1\right)$$

To solve this problem, an expectation-maximum (EM) like algorithm is employed.

Alternate between the two variables, minimizing over one while keeping the other one fixed.



Sparse Coding

Extend the *empirical cost* to the *expected cost*: Bottou and Bousquet (2008)

$$f_n(\mathbf{D}) \triangleq \frac{1}{n}\sum_{i=1}^{n} l(\mathbf{x}_i, \mathbf{D}),$$

$$f(\mathbf{D}) \triangleq \mathbb{E}_{\mathbf{x}}[l(\mathbf{x}, \mathbf{D})] = \lim_{n\to\infty} f_n(\mathbf{D})$$

where the expectation is taken relative to the (unknown) probability distribution p(x) of the data.

Sparse Coding

Calculating dictionary in classical sparse coding

First order stochastic gradient descent: Aharon and Elad (2008)

$$\mathbf{D}_t = \Pi_{\mathcal{C}}\left[\mathbf{D}_{t-1} - \frac{\rho}{t}\nabla_{\mathbf{D}} l(\mathbf{x}_t, \mathbf{D}_{t-1})\right], \qquad (6)$$

where ρ is the gradient step, $\Pi_{\mathcal{C}}$ is the orthogonal projector on $\mathcal{C}$, and the training set $\mathbf{x}_1, \mathbf{x}_2, \ldots$ are i.i.d. samples of the (unknown) distribution p(x).

ρ

Sparse Coding

Online Dictionary Learning:

1. Based on stochastic approximations.
2. Processing one sample at a time.
3. Not requiring explicit learning rate ρ tuning.

Classical first-order stochastic gradient descent

1. Good initialization of ρ .
2. minimizes a sequentially quadratic local approximations of the expected cost.

Sparse Coding

Sparse Coding Step:

computing the decomposition α_t of x_t over the dictionary D_{t-1}

Dictionary Update Step:

D_t is computed by minimizing over $\mathcal{C}$ the function

$$\hat{f}_t(D) \triangleq \frac{1}{t}\sum_{i=1}^{t}\frac{1}{2}||x_i - D\alpha_i||_2^2 + \lambda||\alpha_i||_1,$$

Algorithm 1 Online dictionary learning.

Require: $x \in \mathbb{R}^m \sim p(x)$ (random variable and an algorithm to draw i.i.d samples of p), $\lambda \in \mathbb{R}$ (regularization parameter), $D_0 \in \mathbb{R}^{m\times k}$ (initial dictionary), T (number of iterations).

1: $A_0 \leftarrow 0$, $B_0 \leftarrow 0$ (reset the "past" information).
2: **for** $t = 1$ to T **do**
3: Draw x_t from $p(x)$.
4: Sparse coding: compute using LARS

$$\alpha_t \triangleq \arg\min_{\alpha\in\mathbb{R}^k} \frac{1}{2}||x_t - D_{t-1}\alpha||_2^2 + \lambda||\alpha||_1. \quad (8)$$

5: $A_t \leftarrow A_{t-1} + \alpha_t\alpha_t^T$.
6: $B_t \leftarrow B_{t-1} + x_t\alpha_t^T$.
7: Compute D_t using Algorithm 2, with D_{t-1} as warm restart, so that

$$D_t \triangleq \arg\min_{D\in\mathcal{C}} \frac{1}{t}\sum_{i=1}^{t}\frac{1}{2}||x_i - D\alpha_i||_2^2 + \lambda||\alpha_i||_1,$$
$$= \arg\min_{D\in\mathcal{C}} \frac{1}{t}\left(\frac{1}{2}\mathrm{Tr}(D^T D A_t) - \mathrm{Tr}(D^T B_t)\right). \quad (9)$$

8: **end for**
9: **Return** D_T (learned dictionary).

Sparse Coding

Motivation:

- The quadratic function $\hat{f}_t$ aggregates the past information computed during the previous steps of the algorithm, $\hat{f}_t$ acts as a *surrogate* for f_t.

- Since $\hat{f}_t$ is close to $\hat{f}_{t-1}$, $\mathbf{D}_t$ can be obtained efficiently using $\mathbf{D}_{t-1}$ as warm restart.

Sparse Coding

Due to the convexity of

$$\hat{f}_t(\mathbf{D}) \triangleq \frac{1}{t}\sum_{i=1}^{t}\frac{1}{2}||\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i||_2^2 + \lambda||\boldsymbol{\alpha}_i||_1,$$

dictionary D convergence to a global optimum is guaranteed.

$$\mathbf{D}_t = \Pi_{\mathcal{C}}\left[\mathbf{D}_{t-1} - \frac{\rho}{t}\nabla_{\mathbf{D}} l(\mathbf{x}_t, \mathbf{D}_{t-1})\right],$$

Algorithm 2 Dictionary Update.

Require: $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_k] \in \mathbb{R}^{m\times k}$ (input dictionary),
$\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_k] \in \mathbb{R}^{k\times k} = \sum_{i=1}^{t}\boldsymbol{\alpha}_i\boldsymbol{\alpha}_i^T$,
$\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_k] \in \mathbb{R}^{m\times k} = \sum_{i=1}^{t}\mathbf{x}_i\boldsymbol{\alpha}_i^T$.

1: **repeat**
2: **for** $j = 1$ to k **do**
3: Update the j-th column to optimize for (9):

$$\mathbf{u}_j \leftarrow \frac{1}{\mathbf{A}_{jj}}(\mathbf{b}_j - \mathbf{D}\mathbf{a}_j) + \mathbf{d}_j.$$
$$\mathbf{d}_j \leftarrow \frac{1}{\max(||\mathbf{u}_j||_2, 1)}\mathbf{u}_j. \quad (10)$$

4: **end for**
5: **until convergence**
6: **Return** D (updated dictionary).

Sparse Coding

Key:

$$\min_{\mathbf{D}\in\mathcal{C},\boldsymbol{\alpha}\in\mathbb{R}^{k\times n}} \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{2}||\mathbf{x}_i-\mathbf{D}\boldsymbol{\alpha}_i||_2^2+\lambda||\boldsymbol{\alpha}_i||_1\right)$$

where λ is a regularization parameter.

Adapted Version in Explicit Region Modeling

The explicit region Λ_{ex} of a video $\mathbf{I}$ is decomposed into n_{ex} disjoint domains (usually $n_{ex} = O(10^2)$),

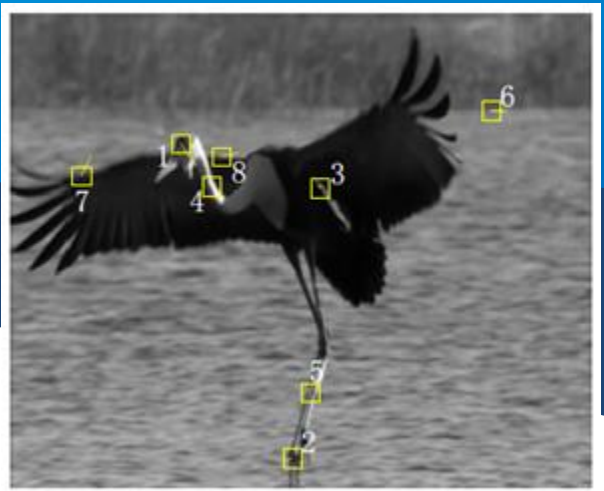
$$\Lambda_{ex} = \bigcup_{i=1}^{n_{ex}} \Lambda_{ex,i}. \tag{3}$$

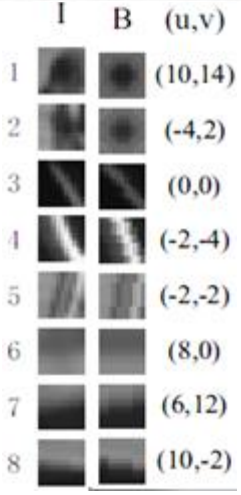
Adapted Version in Explicit Region Modeling

A primitive can be represented by a motion primitive $B_i \in \Delta_B$,

$$\mathbf{I}(x, y, t) = \alpha_i B_i(x, y, t) + \epsilon, \quad \forall (x, y, t) \in \Lambda_{ex,i}. \quad (4)$$

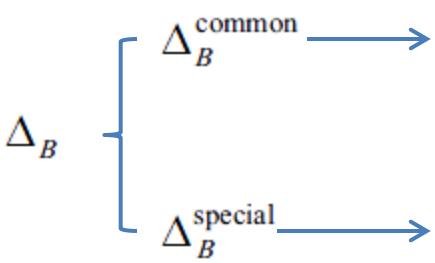
B_i means the ith primitive from the primitive dictionary Δ_B, which fits the brick $\mathbf{I}_{\Lambda_{ex,i}}$ best. Here i indexes the parameters such as type, position, orientation and scale of B_i. α_i is the corresponding coefficient. ϵ represents the residue, which is assumed to be i.i.d. Gaussian.





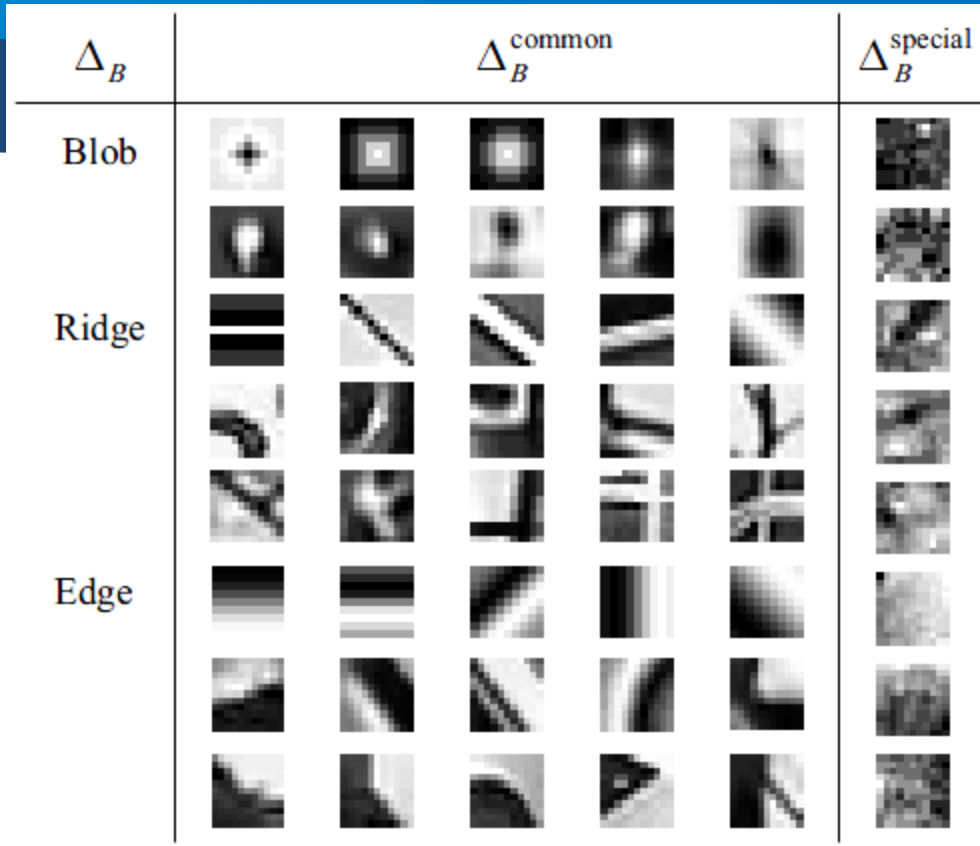
Adapted Version in Explicit Region Modeling

a minority of noisy bricks are trackable over time but not sketchable; thus we cannot find specific shared primitives to represent them.



Trackable and Sketchable Regions

Trackable and Non-sketchable Regions



Adapted Version in Explicit Region Modeling

Based on the representation in eqn(4), the probabilistic model of trackable parts in $\mathbf{I}_{\Lambda_{ex}}$ is defined as

$$p(\mathbf{I}_{\Lambda_{ex}}; \mathbf{B}, \alpha) = \prod_{i=1}^{n_{ex}} \frac{1}{(2\pi)^{\frac{n}{2}} \sigma_i^n} \exp\{-E_i\}$$
$$E_i = \sum_{(x,y,t)\in\Lambda_{ex,i}} \frac{(\mathbf{I}(x,y,t) - \alpha_i B_i(x,y,t))^2}{2\sigma_i^2}. \quad (5)$$

where $\mathbf{B} = (B_1, ..., B_{n_{ex}})$ represents the selected primitive set and $n = |\Lambda_{ex,i}|$.

Adapted Version in Explicit Region Modeling

In order to alleviate computational complexity, α are calculated by filter responses.

The fitted filter F gives a raw sketch of the trackable patch and extracts information. such as type and orientation, for generating the primitive.

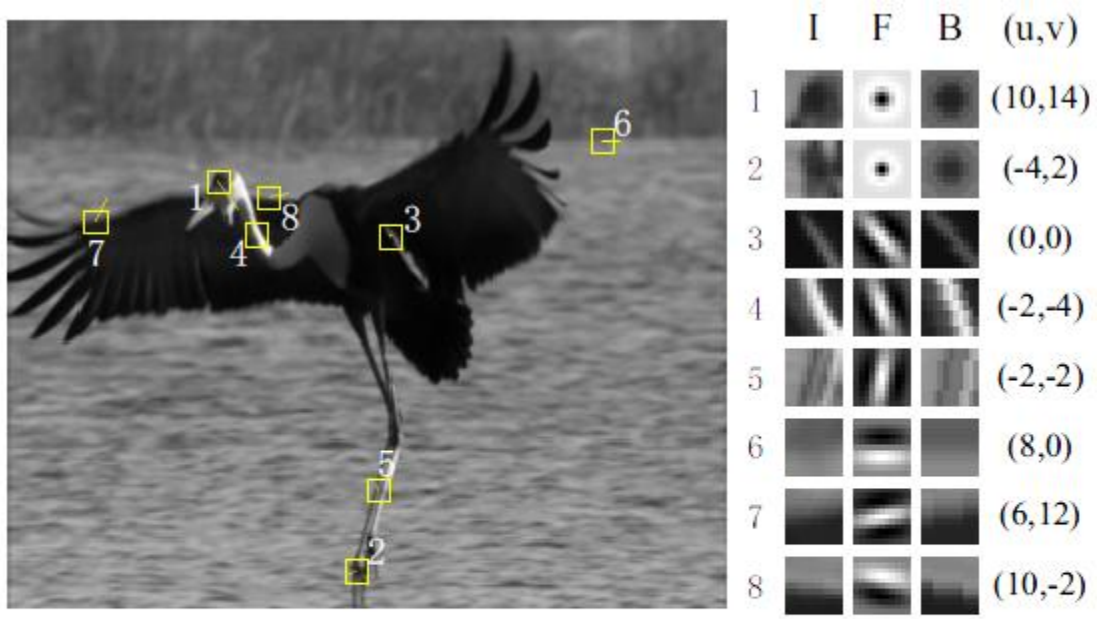


Figure 4. Some examples of primitives in a frame of video. Each group shows the original local image $\mathbf{I}$, the best fitted filter $\mathbf{F}$, the fitted primitive $\mathbf{B} \in \Delta_B$ and the velocity (u, v), which represents the motion of $\mathbf{B}$.

Sketch Pursuit for Primal Sketch

The image lattice Λ is divided into the sketchable and non-sketchable parts for the structural and textural parts respectively.

$$\Lambda = \Lambda_{\text{sk}} \cup \Lambda_{\text{nsk}}, \quad \Lambda_{\text{sk}} \cap \Lambda_{\text{nsk}} = \emptyset. \tag{6}$$

The sketchable part is further divided into a number of disjoint patches with each patch being fitted by an image primitive.

$$\Lambda_{\text{sk}} = \cup_{k=1}^{K} \Lambda_{\text{sk},k}, \quad \Lambda_{\text{sk},k_1} \cap \Lambda_{\text{sk},k_2} = \emptyset, k_1 \neq k_2. \tag{7}$$

Sketch Pursuit for Primal Sketch

The selected image primitives is indexed by $k = 1, 2, \ldots, K$,

$$k = (\theta_{\text{topological}}, \theta_{\text{geometric}}, \theta_{\text{photometric}}), \tag{8}$$

where $\theta_{\text{topological}}$ is the type (degree of arms) of the primitive (blob, terminator, corner, junctions etc), $\theta_{\text{geometric}}$ collects the locations of the landmarks of the primitive, and $\theta_{\text{photometric}}$ collects the intensity profiles of the arms of the primitive.

Sketch Pursuit for Primal Sketch

The sketch graph is a layer of hidden representation which has to be inferred from the image,

$$S_{\text{sk}} = (K, (\Lambda_{\text{sk},k}, B_k, a_k), k = 1, 2, ..., K),$$

where S_{sk} decides the sketchable part of the image, B_k is the image patch for primitive k, and a_k is the address variable pointing to the neighbors of the vertex $S_{\text{sk},k} = (\Lambda_{\text{sk},k}, B_k)$. We adopt the following generative image model on Λ_{sk}

$$\mathbf{I}_{\Lambda_{\text{sk}},k} = B_k + \mathbf{n}, \quad k = 1, 2, ..., K. \tag{9}$$

Sketch Pursuit for Primal Sketch

Probability model for the primal sketch representation:

$$
\begin{aligned}
& p(\mathbf{I}_\Lambda, S_{\text{sk}}, S_{\text{nsk}}) \\
= \; & \frac{1}{Z} \exp\{ - \frac{1}{2\sigma_o^2} \sum_{k=1}^{K} \sum_{(u,v)\in\Lambda_{\text{sk},k}} (\mathbf{I}(u,v) - B_k(u,v))^2 - \sum_{m=1}^{M} \sum_{i=1}^{n} \langle \beta_{mi}, h_i(\mathbf{I}_{\Lambda_{\text{nsk},m}}) \rangle \\
& - E(S_{\text{sk}}) - E(S_{\text{nsk}}) \}, \qquad (13)
\end{aligned}
$$

Sparse Coding Residual Error

FRAME Residual Error

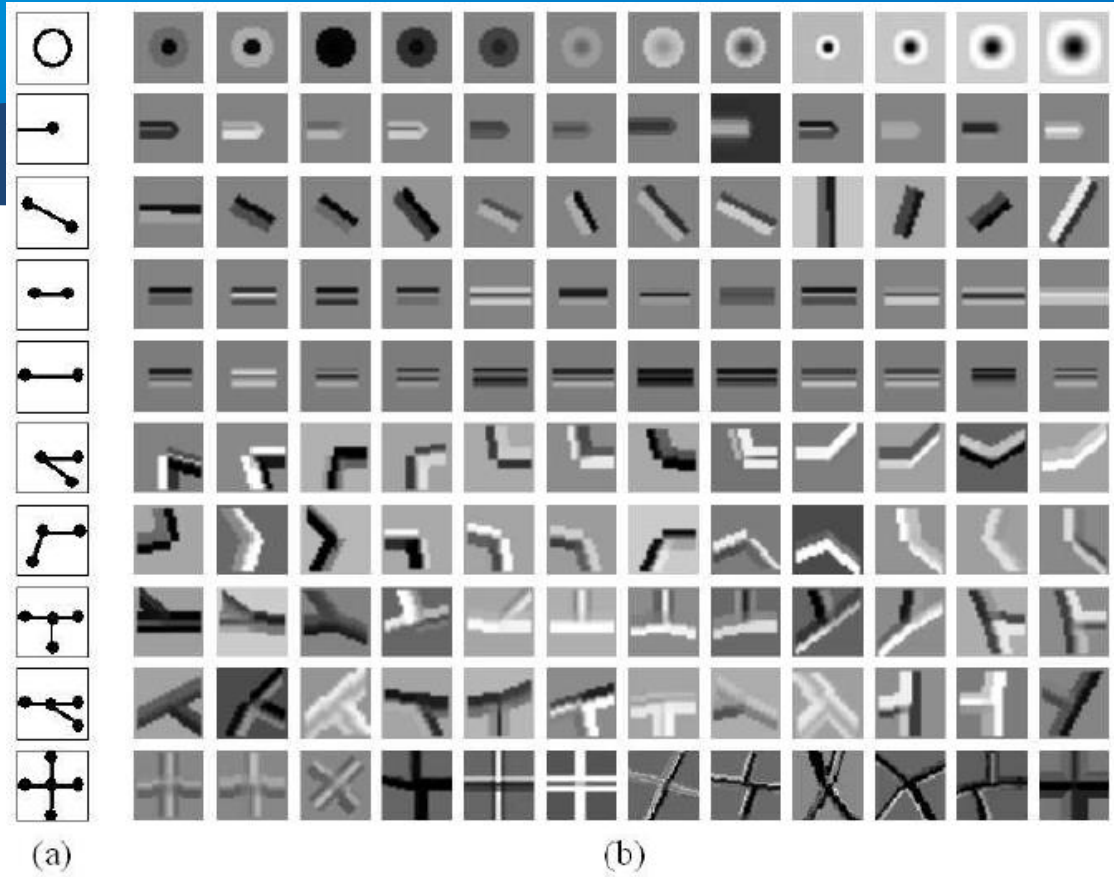
Dictionary Coding Length

FRAME Coding Length

Sketch Pursuit for Primal Sketch

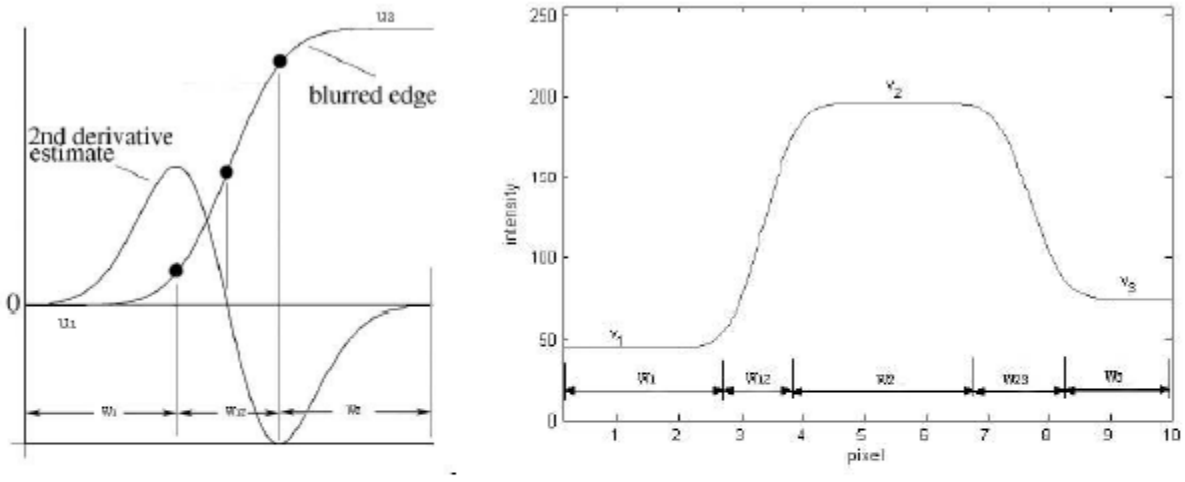
The dictionary of image primitives designed for the sketch graph *Ssk* consists of eight types of primitives in increasing degree of connection:

0. blob.
1. terminators, edge, ridge.
2. multi-ridge, corner.
3. junction.
4. cross.



Sketch Pursuit for Primal Sketch

These primitives have a center landmark and $l = 0$ ~ 4 axes (arms) for connecting with other primitives. For arms, the photometric property is represented by the intensity profiles.

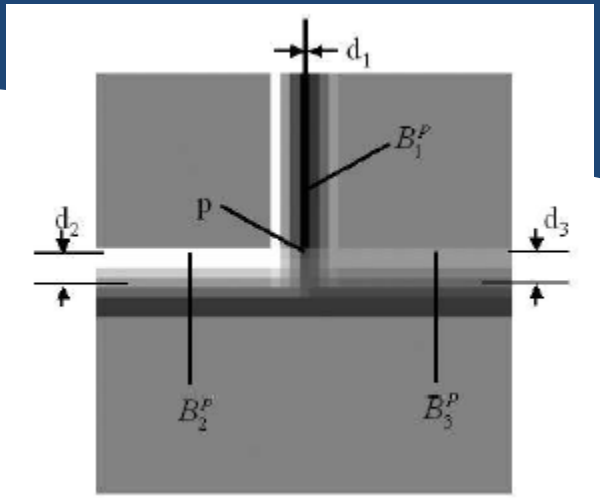


(a) Edge profile (b) Ridge profile

Figure 8: (a) The edge profile is represented by 5 parameters. The illustration of the computing of the scale for a blurred edge. The blurring scale is measured by the distance between the extremes of the second derivative. (b) The representation of a ridge profile with 8 parameters.

Sketch Pursuit for Primal Sketch

For the center of a primitive, considering the arms may overlap with each other, a pixel *p* with *L* arms overlapped is modeled by:



$$B_k(p) = \frac{1}{D}\sum_{l=1}^{L}\frac{A_l^p}{d_l+1},$$

where $D = \sum_{l=1}^{L}\frac{1}{d_l+1}$. the intensities of the profiles at this pixel p as A_1^p, A_2^p, ..., A_L^p,

the distances from the point p to the center lines of these arms as d_1, d_2, ..., d_L

Sketch Pursuit for Primal Sketch

divide the set of vertices V into 5 subsets according to their degrees of connection,

$$V = V_0 \cup V_1 \cup V_2 \cup V_3 \cup V_4, \tag{15}$$

where V_i is the set of vertices with degree i. Then we have

$$E(S_{\text{sk}}) = \sum_{d=0}^{4} \lambda_d |V_d|, \tag{16}$$

where $|V_d|$ is the cardinality of the set V_d, and λ_d can be interpreted as the coding length associated with each types of vertices. $\lambda_0 = 1.0$, $\lambda_1 = 5.0$, $\lambda_2 = 2.0$, $\lambda_3 = 3.0$,$\lambda_4 = 4.0$.

According to Gestalt laws, the closure and continuity are preferred in the perceptual organization. Thus we penalize terminators, edges, ridge.

Sketch Pursuit for Primal Sketch

The Sketch Pursuit Algorithm consists of two phases:

Phase 1: Deterministic pursuit of the sketch graph S_{sk} in a procedure similar to matching pursuit. It sequentially add new strokes (primitives of edges/ridges) that are most prominent.

Phase 2: Refine the sketch graph S_{sk} to achieve better Gestalt organization by reversible graph operators, in a process of maximizing a posterior probability (MAP).

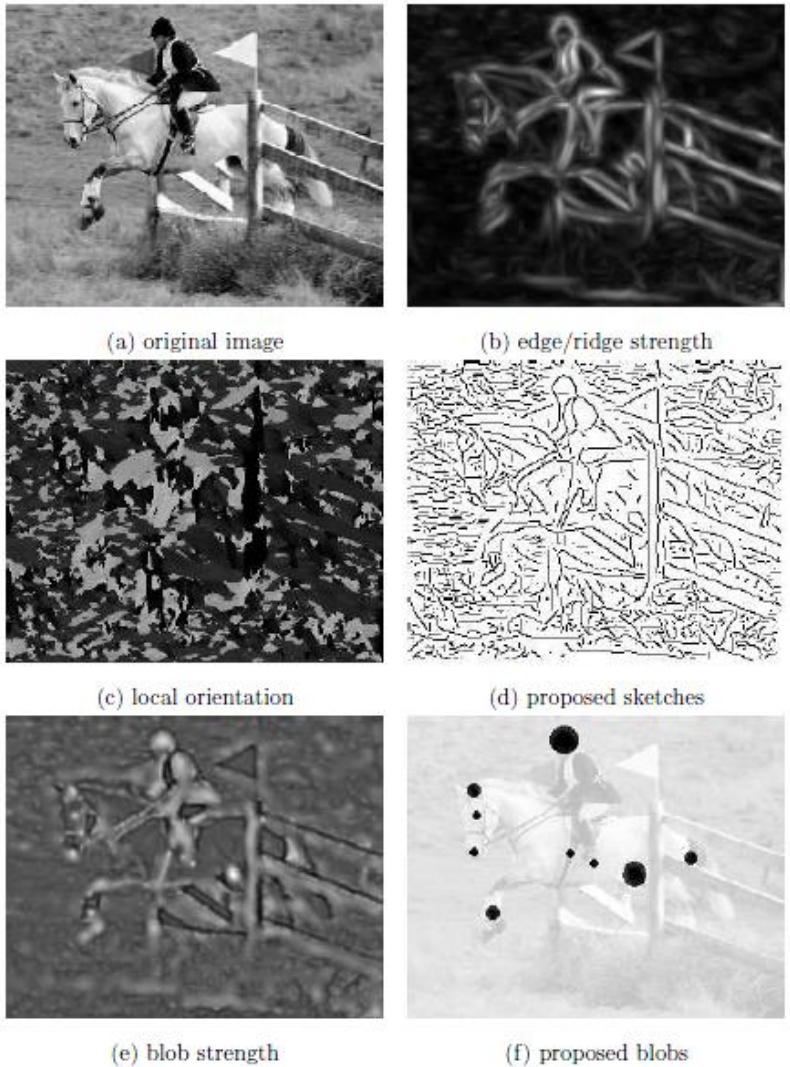
Coarse to Fine

Sketch Pursuit for Primal Sketch

Phase 1

Blob-Edge-Ridge (BER) Detector for a proposal map S_{sk}^{∞}

Acting as a prior for sketch pursuit algorithm.



(a) original image (b) edge/ridge strength (c) local orientation (d) proposed sketches (e) blob strength (f) proposed blobs

Sketch Pursuit for Primal Sketch

Phase 1

This operation is called *creation* and defined as graph operator O_1.
The reverse operation O'_1 proposes to remove one stroke.

operators	graph change	illustration
O_1, O'_1	create / remove a stroke	

Sketch Pursuit for Primal Sketch

Phase 1

This operation is called *growing* and defined as graph operator $O2$.
This operator can be applied iteratively until no proposal is accepted.
Then a curve is obtained.

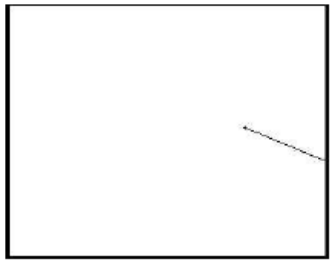
O_2, O_2'	grow / shrink a stroke	

Sketch Pursuit for Primal Sketch

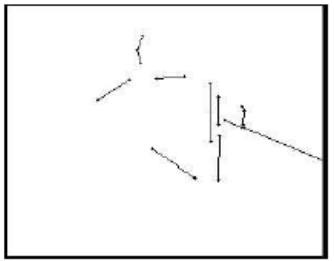
Phase 1

The sketch pursuit phase I applies operators $O1$ and $O2$ iteratively until no more strokes are accepted.

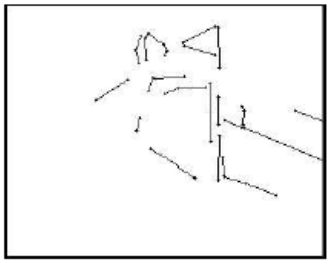
Phase I provides an initialization state for sketch pursuit phase II.



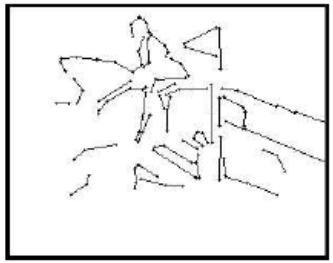
(a) iteration 1



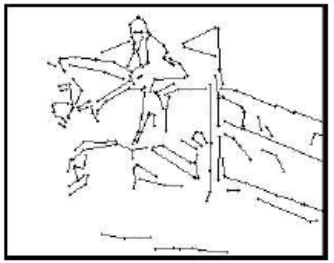
(b) iteration 10



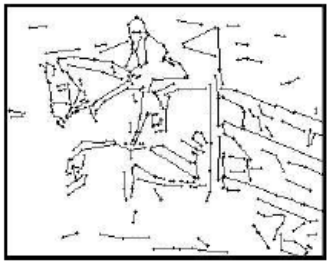
(c) iteration 20



(d) iteration 50



(e) iteration 100



(f) iteration 180

Sketch Pursuit for Primal Sketch

Probability model for the primal sketch representation:

$$
\begin{aligned}
& p(\mathbf{I}_\Lambda, S_{\text{sk}}, S_{\text{nsk}}) \\
= \; & \frac{1}{Z} \exp\{ - \frac{1}{2\sigma_o^2} \sum_{k=1}^{K} \sum_{(u,v)\in\Lambda_{\text{sk},k}} (\mathbf{I}(u,v) - B_k(u,v))^2 - \sum_{m=1}^{M} \sum_{i=1}^{n} \langle \beta_{mi}, h_i(\mathbf{I}_{\Lambda_{\text{nsk},m}}) \rangle \\
& - E(S_{\text{sk}}) - E(S_{\text{nsk}}) \}, \qquad (13)
\end{aligned}
$$

Sparse Coding Residual Error

FRAME Residual Error

Dictionary Coding Length

FRAME Coding Length

Sketch Pursuit for Primal Sketch

Phase 1

Using a simplified primal sketch model

Sparse Coding Residual Error

$$p_I(\mathbf{I}_\Lambda, S_{\text{sk}}, S_{\text{nsk}}; \Delta_{\text{sk}}) \approx \frac{1}{Z} \exp\{-\frac{1}{2\sigma_o^2} \left(\sum_{k=1}^{K} \sum_{(u,v)\in\Lambda_{\text{sk},k}} (\mathbf{I}(u,v) - B_k(u,v))^2 + \sum_{(u,v)\in\Lambda_{\text{nsk}}} (\mathbf{I}(u,v) - \mu(u,v))^2 \right)\},$$

$\mu(u,v)$ is the local intensity mean around pixel (u,v).

Simplify FRAME Residual Error as a local Gaussian distribution.

Sketch Pursuit for Primal Sketch

Phase 1

we add a stroke $S_{\text{sk},K+1}$ into S_{sk}, then $S'_{\text{sk}} = S_{\text{sk}} \cup S_{\text{sk},K+1}$, $\Lambda'_{\text{nsk}} = \Lambda_{\text{nsk}} - \Lambda_{\text{sk},K+1}$, and the probability changes to

$$p_I(\mathbf{I}_\Lambda, S'_{\text{sk}}, S'_{\text{nsk}}; \Delta_{\text{sk}}) \approx \frac{1}{Z} \exp\{-\frac{1}{2\sigma_o^2} \quad (\sum_{k=1}^{K+1} \sum_{(u,v)\in\Lambda_{\text{sk},k}} (\mathbf{I}(u,v) - B_k(u,v))^2 + \sum_{(u,v)\in\Lambda'_{\text{nsk}}} (\mathbf{I}(u,v) - \mu(u,v))^2)\}, \tag{19}$$

Comparing (19) and (18), define

$$\begin{aligned} \Delta\mathcal{L} &= \log \frac{p(\mathbf{I}_\Lambda, S'_{\text{sk}}, S'_{\text{nsk}}; \Delta_{\text{sk}})}{p(\mathbf{I}_\Lambda, S_{\text{sk}}, S_{\text{nsk}}; \Delta_{\text{sk}})} \\ &= \frac{1}{2\sigma_o^2}\{ \sum_{(u,v)\in\Lambda_{\text{sk},K+1}} (\mathbf{I}(u,v) - \mu(u,v))^2 - (\mathbf{I}(u,v) - B_{(K+1)}(u,v))^2\}, \end{aligned} \tag{20}$$

which is called *image coding length gain* by adding a new stroke.

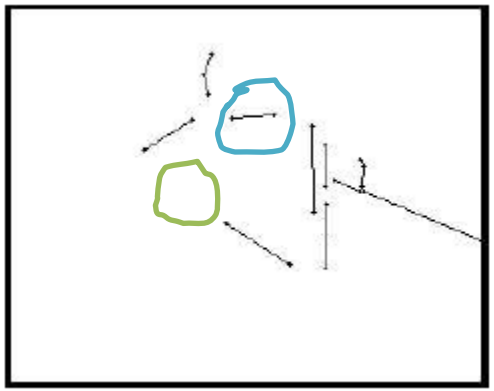
Sketch Pursuit for Primal Sketch

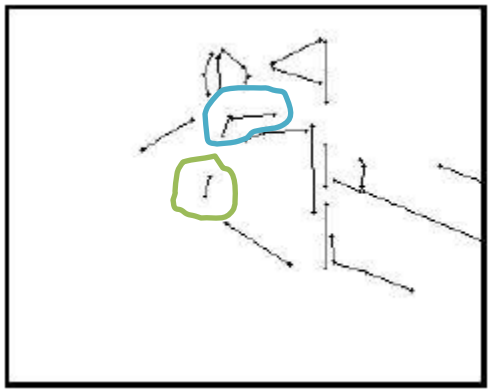
Phase 1

From each end of the accepted stroke $S_{\text{sk},K}$, we search the connected points in S_{sk}^{∞} until a linelet is formed within a pre-specified average fitting error per pixel. We denote it as a new proposed stroke $S_{\text{sk},K+1}$.

Grow a stroke

Grow a stroke





Sketch Pursuit for Primal Sketch

Phase 2

Sketch Pursuit by Reversible Graph Operators In the sketch pursuit phase II, the sketch graph S_{sk} is refined to achieve better Gestalt organization by the ten pairs of the reversible graph operators discussed above, in a process of maximizing a posterior (MAP).

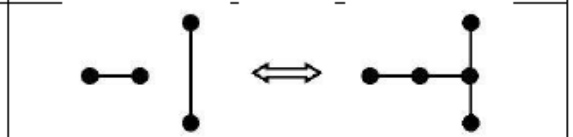
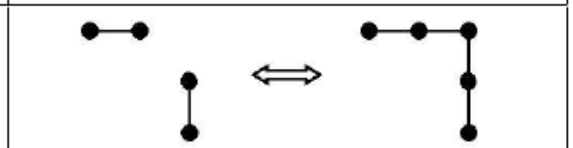
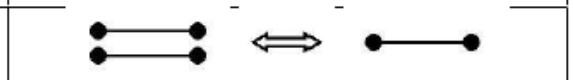
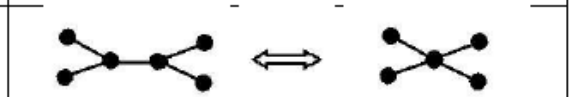
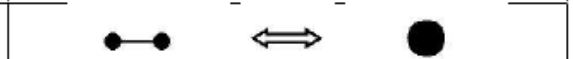
$$(S_{\text{sk}}, S_{\text{nsk}})^* = \arg\max p_{II}(S_{\text{sk}}, S_{\text{nsk}} | \mathbf{I}_\Lambda; \Delta_{\text{sk}})$$

Sketch Pursuit for Primal Sketch

Phase 2

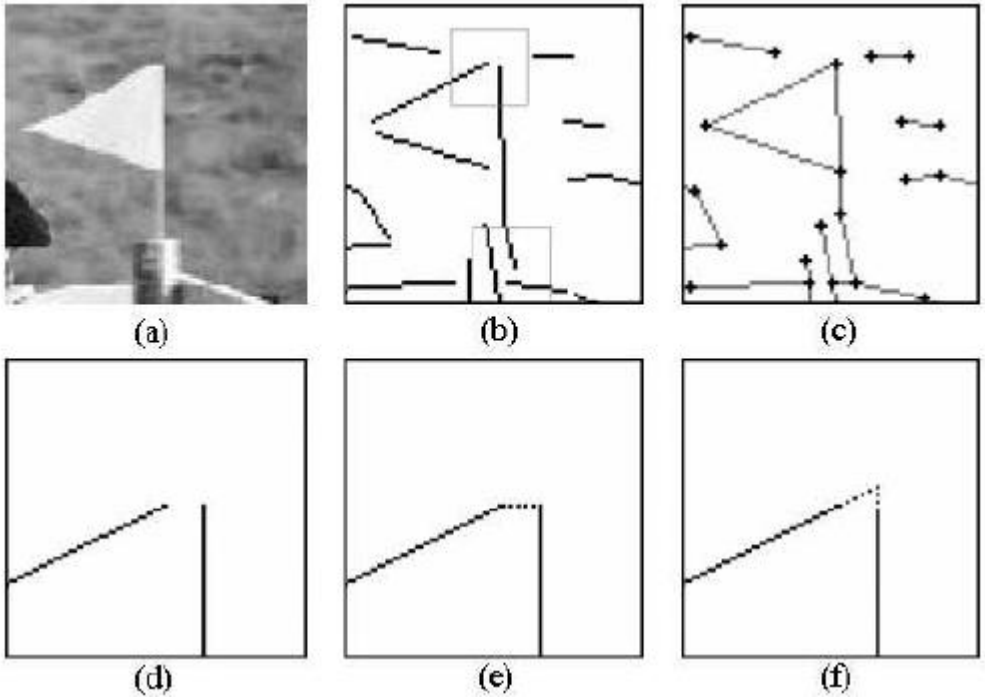
Overall 10 graph operators is proposed facilitate the sketch pursuit process to transverse the sketch graph space.

Simplified Version of DDMCMC

O_3, O_3'	connect / disconnect vertices	
O_4, O_4'	extend one stroke and cross / disconnect and combine	
O_5, O_5'	extend two strokes and cross / disconnect and combine	
O_6, O_6'	combine two connected strokes / break a stroke	
O_7, O_7'	combine two parallel strokes / split one into two parallel	
O_8, O_8'	merge two vertices / split a vertex	
O_9, O_9'	create / remove a blob	
O_{10}, O_{10}'	switch between a stroke(s) and a blob	

Sketch Pursuit for Primal Sketch

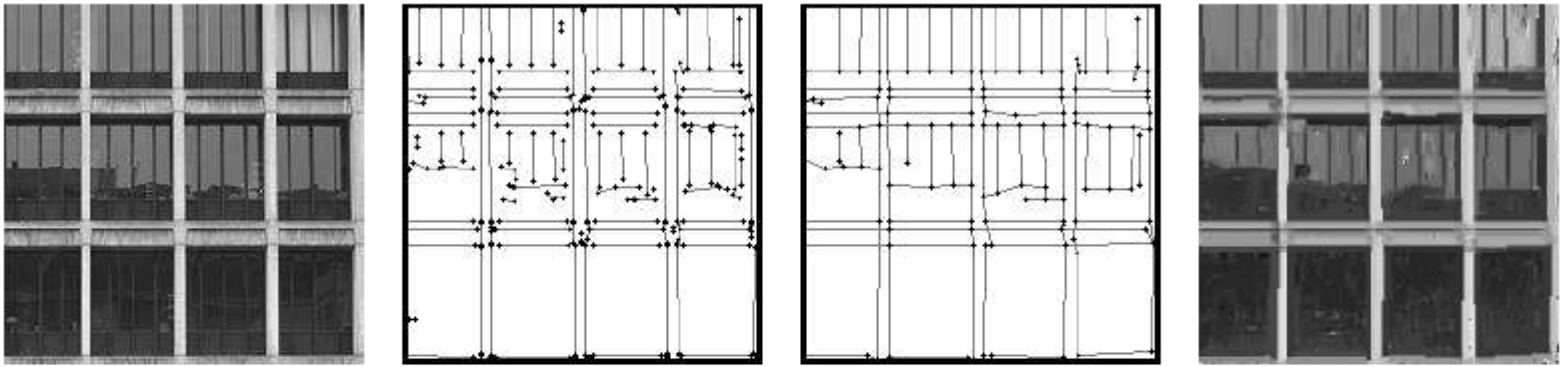
Phase 2



a. Input image.
b. Sketch map after Phase 1.
c. Sketch map after Phase 2.
d. The zoom-in view of the upper rectangle in b.
e. Applying O3 – connecting two vertices.
f. Applying O5 – extending two strokes and cross.

Sketch Pursuit for Primal Sketch

Phase 2



(a) (b) (c) (d)

Figure 13: One result of the primal sketch model. (a) input image; (b) raw sketch graph after sketch pursuit phase I; (c) final sketch graph after sketch pursuit phase II; (d) reconstructed image from our primal sketch model.

Sketch Pursuit for Primal Sketch

Probability model for the primal sketch representation:

$$
\begin{aligned}
& p(\mathbf{I}_\Lambda, S_{\text{sk}}, S_{\text{nsk}}) \\
= \; & \frac{1}{Z} \exp\{ - \frac{1}{2\sigma_o^2} \sum_{k=1}^{K} \sum_{(u,v)\in\Lambda_{\text{sk},k}} (\mathbf{I}(u,v) - B_k(u,v))^2 - \sum_{m=1}^{M} \sum_{i=1}^{n} \langle \beta_{mi}, h_i(\mathbf{I}_{\Lambda_{\text{nsk},m}}) \rangle \\
& - E(S_{\text{sk}}) - E(S_{\text{nsk}}) \}, \qquad (13)
\end{aligned}
$$

Sparse Coding Residual Error

FRAME Residual Error

Dictionary Coding Length

FRAME Coding Length

Sketch Pursuit for Primal Sketch

Phase 2

From the initialization result of the sketch pursuit phase I, by applying a set of graph operators, the sketch pursuit phase II maximizes a simplified version of the joint probability (13).

$$p_{II}(\mathbf{I}_\Lambda, S_{\text{sk}}, S_{\text{nsk}}; \Delta_{\text{sk}}) \approx \frac{1}{Z} \exp\{-\frac{1}{2\sigma_o^2} \Big(\sum_{k=1}^{K} \sum_{(u,v)\in\Lambda_{\text{sk},k}} (\mathbf{I}(u,v) - B_k(u,v))^2 + \sum_{(u,v)\in\Lambda_{\text{nsk}}} (\mathbf{I}(u,v) - \mu(u,v))^2 \Big) - E(S_{\text{sk}})\}, \quad (21)$$

Sparse Coding Residual Error

Simplify FRAME Residual Error as a local Gaussian distribution.

Dictionary Coding Length

Sketch Pursuit for Primal Sketch

Phase 2

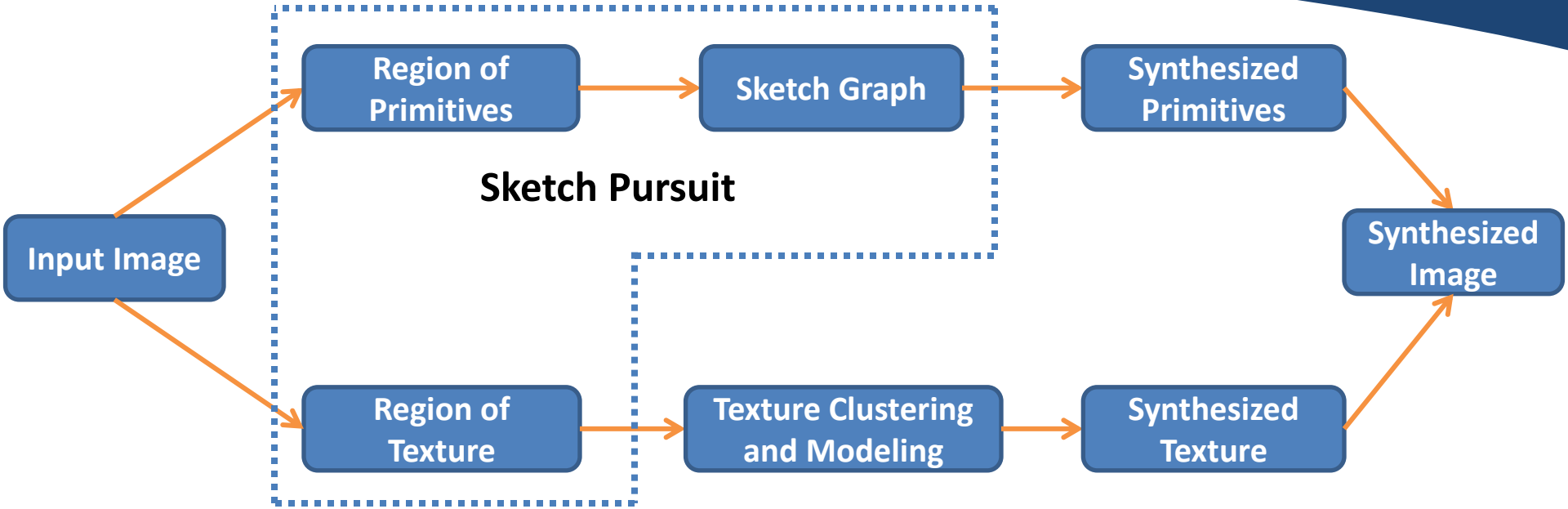
$$
\begin{aligned}
(S_{\text{sk}}, S_{\text{nsk}})^* &= \arg\max p_{II}(S_{\text{sk}}, S_{\text{nsk}} | \mathbf{I}_\Lambda; \Delta_{\text{sk}}) && (22)\\
&= \arg\max p_{II}(\mathbf{I}_\Lambda, S_{\text{sk}}, S_{\text{nsk}}; \Delta_{\text{sk}}) && (23)\\
&= \arg\min \frac{1}{2\sigma_o^2}\Big(\sum_{k=1}^{K}\sum_{(u,v)\in\Lambda_{\text{sk},k}} (\mathbf{I}(u,v) - B_k(u,v))^2\\
&+ \sum_{(u,v)\in\Lambda_{\text{nsk}}} (\mathbf{I}(u,v) - \mu(u,v))^2\Big) + E(S_{\text{sk}})\\
&= \arg\min \mathcal{L}_A + \mathcal{L}_S\\
&= \arg\min \mathcal{L}(S_{\text{sk}}, S_{\text{nsk}}) && (24)
\end{aligned}
$$

where $\mathcal{L}_A = \frac{1}{2\sigma_o^2}\big(\sum_{k=1}^{K}\sum_{(u,v)\in\Lambda_{\text{sk},k}}(\mathbf{I}(u,v) - B_k(u,v))^2 + \sum_{(u,v)\in\Lambda_{\text{nsk}}}(\mathbf{I}(u,v) - \mu(u,v))^2\big)$ is called image coding length, $\mathcal{L}_S = E(S_{\text{sk}})$ is called sketch coding length, $\mathcal{L}(S_{\text{sk}}, S_{\text{nsk}}) = \mathcal{L}_A + \mathcal{L}_S$ is called total coding length.

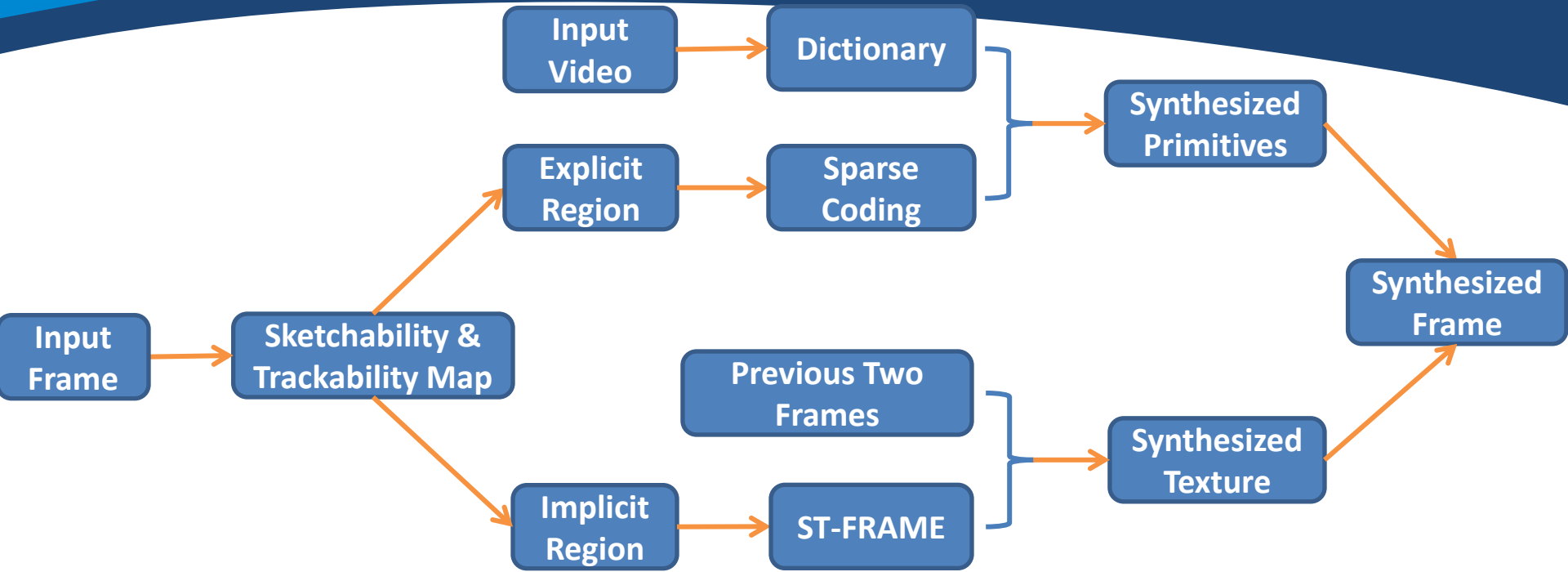
Episode 4

Reviews, Problems, Explanations, and Vista

The Framework of Primal Sketch



The Review of Video Primal Sketch



Problem in Video Primal Sketch

(a) Input

Video Resolution	288×352 pixels
Explicit Region	31,644 pixels≈ 30%
Primitive Number	300
Primitive Width	11 pixels
Explicit Parameters	3,600 ≈ 3.6%
Implicit parameters	15×(11+12+5)=420

Table 1. The parameters in video primal sketch model for the water bird video in Fig.2

Major region: implicit region
Major model parameters: explicit parameters

Problem in Video Primal Sketch

(a) Input

Example	Size(Pixels)	Error($I_{\Lambda_{ex}}$)	Error($I_{\Lambda_{im}}$)
1	190×330	5.37%	0.59%
2	288×352	3.07%	0.16%
3	288×352	2.8%	0.17%

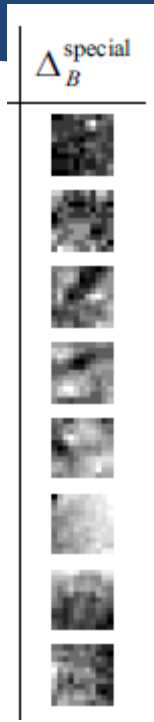
Table 4. Error assessment of synthesized videos.

Major error: error from reconstructing explicit regions

Problem in Video Primal Sketch

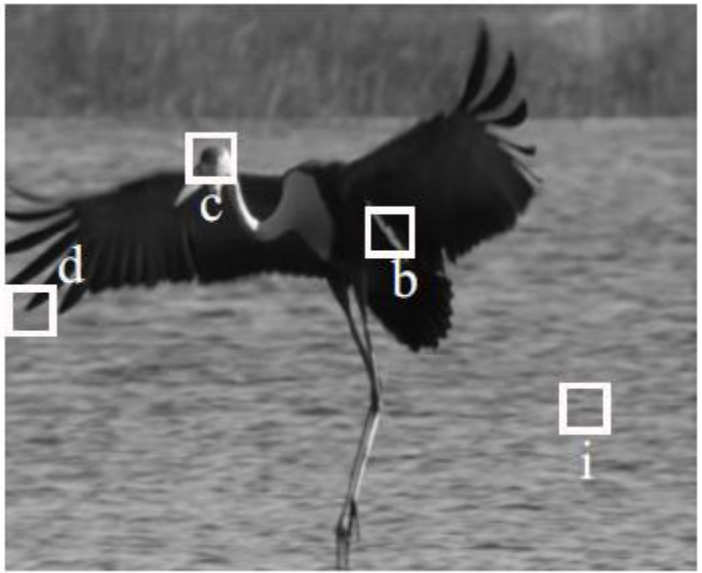
(a) Input

Special dictionary for trackable and non-sketchable region.



Modeling trackable and non-sketchable region with Sparse Coding or FRAME ?

Problem in Video Primal Sketch



	Sketchable	Non-sketchable
Trackable	(a) Moving Edge (b) Moving Bar (c) Moving Blob (d) Moving Corner	(g) Moving Kernel
Intrackable	(e) High-speed Moving Edge (f) High-speed Moving Bar	(h) Flat Area (i) Textured Motion

Problem in Both Methods

Probability model for the primal sketch representation:

$$
\begin{aligned}
& p(\mathbf{I}_\Lambda, S_{\text{sk}}, S_{\text{nsk}}) \\
= \; & \frac{1}{Z}\exp\{-\frac{1}{2\sigma_o^2}\sum_{k=1}^{K}\sum_{(u,v)\in\Lambda_{\text{sk},k}}(\mathbf{I}(u,v)-B_k(u,v))^2 - \sum_{m=1}^{M}\sum_{i=1}^{n}\langle\beta_{mi}, h_i(\mathbf{I}_{\Lambda_{\text{nsk},m}})\rangle \\
& -E(S_{\text{sk}}) - E(S_{\text{nsk}})\},
\end{aligned}
$$

Simplified as $\sum_{(u,v)\in\Lambda_{\text{nsk},m}} (h(u,v)-h_m)^T\Sigma_m^{-1}(h(u,v)-h_m)$

Problem in Methods

Probability model for the video primal sketch representation:

$$p(\mathbf{I}|\mathbf{B},\mathbf{F},\alpha,\beta) = \tag{12}$$

$$\frac{1}{Z}\exp\{-\sum_{i=1}^{n_{ex}}\sum_{(x,y,t)\in\Lambda_{ex,i}}\frac{(\mathbf{I}(x,y,t)-\alpha_i B_i(x,y,t))^2}{2\sigma_i^2}$$

$$-\sum_{j=1}^{n_{im}}\sum_{k=1}^{K}\langle\beta_{k,j},H_k(\mathbf{I}_{\Lambda_{im,j}}|\mathbf{I}_{\partial\Lambda_{im,j}})\rangle\}.$$

inconsistent energy measurement!

Explanations - Contrary vs. Uniform

S. C. Zhu
"Eternal Debate"

The Collapse of Classical Physics

1. The central problems of primal sketch & video primal sketch:

The great complexity caused by mixing two totally irrelevant model together.

2. Reviewing two method in a dialectic way.

The problem caused by metaphysics: constrained observation, huge gap between two categories.

a. 相对论排除了绝对时空观的牛顿幻觉，
b. 量子论排除了可控测量过程中的牛顿迷梦，
c. 混沌论则排除了拉普拉斯可预见性的狂想.

Vista

3. The philosophical purpose of image / video segmentation:

Magnifying the difference among different parts of the image / video.

4. Complement method to ameliorate these two modeling method

Intuition: particle wave duality, texture & texton, coexist for each atom in image / video, observation decides which state dominates.

Vista

5. Schrödinger Equation / Uncertain Principle:

The particle position we observe is the integral of a probability wave.

6. The new intuition of video modeling

Texton texture duality: (1). Integral of a single probability wave – trackable, sketchable motion, (2). Integral of the composition of several probability wave – dynamic texture

QUESTIONS?