TELECOM ParisTech

Parsing images with finite state machines for object class segmentation and annotation

Automates à états finis stochastiques pour l'annotation et la segmentation d'images en classes d'objets

Hichem Sahbi

2013D003

juin 2013

Département Traitement du Signal et des Images
Groupe TII : Traitement et Interprétation des Images

Parsing Images with Finite State Machines for Object Class Segmentation and Annotation

Hichem SAHBI
CNRS LTCI ; Télécom ParisTech
46 rue Barrault, 75013 Paris
hichem.sahbi@telecom-paristech.fr

Abstract

We introduce in this work a stochastic inference process, for scene annotation and object class segmentation, based on finite state machines (FSMs). The design principle of our framework is generative and based on building, for a given scene, finite state machines that encode annotation lattices, and inference consists in finding and scoring the best configurations in these lattices.
Different operations are defined using our FSM framework including reordering, segmentation, visual transduction, and label-language modeling. All these operations are combined together in order to achieve annotation as well as object class segmentation.

Keywords: Finite State Machines, Statistical Machine Learning, Object Class Segmentation, Image Annotation.

Automates à États Finis Stochastiques pour l’Annotation et la Segmentation d’Images en Classes d’Objets

Résumé

On introduit dans cette contribution un processus d’inférence stochastique basé sur les automates à états finis pour l’annotation et la segmentation de scènes en classes d’objets. La méthode proposée est générative et permet de construire pour une scène donnée des automates à états finis codant les treillis de l’ensemble des annotations possibles de cette scène ainsi que leurs scores.

Différentes opérations sont définies à travers des automates incluant le réordonnancement, la segmentation, la transduction visuelle et la modélisation du langage des labels. Toutes ces opérations sont combinées afin de réaliser le processus global d’annotation et segmentation en classes d’objets.

1 Introduction

The general problem of image annotation is usually converted into classification. Many existing state of the art methods (see for instance [5, 19, 14, 34, 30, 24, 25, 19, 5, 7, 15, 22, 18, 11, 21]) treat each keyword (also referred to as category or concept) as a class, and then build the corresponding concept-specific classifier in order to identify images or image regions belonging to that class, using a variety of machine learning techniques such as Markov models [19, 24], latent Dirichlet allocation [1], probabilistic latent semantic analysis [23], support vector machines [13, 24, 29], etc. Annotation methods may also be categorized into: *region-based* requiring a preliminary step of image segmentation ([20, 33, 17, 26, 28, 32], etc.), and *holistic* ([15, 36], etc.) operating directly on the entire image space. In both cases, training is achieved in order to learn how to attach concepts with the corresponding visual features.

Object class segmentation (OCS) is particularly challenging and has many potential applications including object recognition and image retrieval. Most of the OCS methods are considered as labelling problems working either on individual pixels or on constellations of spatially homogeneous ones, referred to as *superpixels*. Accordingly, state of the art methods may be categorized depending on their partitioning strategies; some of them perform label prediction directly at the pixel level [33] (as the finest partitioning), while others rely on superpixels or preexisting segmentations [2, 12, 38, 32, 17], and variants of them use grouping and intersection [27, 26]. A key issue of OCS is how to model dependencies between pixels, superpixels or objects. In general, dependencies are defined as unary and (possibly high order) interaction potential functions. The former measures the likelihood of a pixel belonging to a particular class, while the latter encodes the dependency information which enforces label consistency or geometrical relationships among neighboring pixels or objects. Dependencies or relationships, in the space of the image in particular, are crucial and they are intuitively modeled by graphs. Such graphs, created at different levels of the processing, may encode the spatial relationship between neighboring points in the image (local geometry, etc.), or between different parts/components of an object (to model the topology or the shape of the object), or between different objects observed in a scene to take into account the context during the annotation process. Recent advances in these aspects usually rely on graphical models, mainly Conditional Random Fields and Markov Random Fields [10], in order to model unary and interaction potentials and learn the conditional distribution over the classes. The most widely used interaction potential function is formulated as a pairwise one [33] and higher-order Potts models [16]. Other related techniques apply logic and stochastic grammars; for a survey of these methods see [39, 35, 31, 6, 37] and references within for detailed discussions.

In this paper, we introduce an original object class segmentation method based on finite state machines (FSMs). Our OCS approach is Bayesian; it finds superpixel labels that maximize a posterior probability, *but* its key-novelty resides in the representational power of FSMs in order to build a comprehensive model for scene segmentation and labeling. Indeed, we translate our OCS into searching implicitly, via FSMs, the optimum of a discrete energy function mixing i) a standard *unary term* that models conditional probability of visual features given their (possible) labels, ii) a standard *interaction potential* (context prior) which provides joint statistics, of co-occurrence, between those labels, and iii) a novel *reordering and grouping term*. The latter allows us, via FSMs, to examine (generate, label, score and rank) many partitions of segments in a given scene, and to return only the most likely ones.

At least three reasons drove us to apply FSMs for object class segmentation:

-Firstly, as FSMs can model huge (even infinite) languages[1], with finite memory and time resources, our method does not require explicit generation of all possible segmentations and labelings. Instead, it first models them implicitly by combining (composing) different FSMs and then efficiently finds the shortest path (i.e., the most likely segmentation and labeling) in a global FSM. Moreover, sparse statistics about unary and interaction potentials allow us to further simplify the individual FSMs and this makes finding the shortest path in the global FSM even faster while maintaining highly

[1] The alphabet of this language corresponds to all the superpixels of a given scene.

effective labeling process (see Section 3).

-Secondly, superpixel reordering allows us to examine all the possible segmentations at different orders and makes it possible, using the chain rule, to maximize the interaction potentials resulting into better labeling performance. For that purpose, scenes are first described with graphs where nodes correspond to superpixels and edges connect neighboring superpixels. Then, reordering is achieved by generating random (Hamiltonian) walks on these graphs using FSMs. Note that graphs with very low connectivity (≤ 4 immediate neighbors for each superpixel) dramatically reduce the complexity of this reordering and also grouping (again see Section 3).

-Finally, the number of possible labels and the non-uniqueness of the solution of our energy function makes it difficult to solve. Indeed, only sub-optimal solutions can be found if standard optimization techniques, such as belief propagation or graph-cut [4, 3], are applied. Instead, FSMs explore larger sets of possible solutions resulting into a more effective and also efficient scene labeling machinery as discussed later in this paper.

The remainder of this paper is organized as follows, In Section 2, we describe our Bayesian model and how to learn its statistics. In Section 3, we describe our main contribution; the inference model based on finite state machines and the integration of those statistics in object class segmentation.

2 Scene Labeling Model

Consider $\mathcal{X}$ as the union of all the possible images of the world. Given n lattice points $\mathcal{V} = \{1, \dots, n\}$, we define $\mathbf{X} = \{X_1, \dots, X_n\} \subset \mathcal{X}$ as a set of *observed* random variables, corresponding to a subdivision of $\mathbf{X}$ into smaller units, referred to as *superpixels*. Let $\mathbf{C} = \{c_i : c_i \subseteq \mathcal{V}\}_{i=1}^{\mathbf{k}}$ be a *random partition* of $\mathcal{V}$ (i.e., $\forall i$, $\forall j \neq i$, $c_i \neq \emptyset$, $c_i \cap c_j = \emptyset$ and $\cup_i\ c_i = \mathcal{V}$); an element $\mathbf{X}_{c_i} \subseteq \mathbf{X}$ is defined as a collection of conditionally independent random variables, i.e., $\mathbf{X}_{c_i} = \{X_k \in \mathbf{X} : k \in c_i\}$. For each partition $\{c_i \subseteq \mathcal{V}\}_i$, we define a set of random variables $\mathbf{Y} = \{\mathbf{Y}_{c_1}, \dots, \mathbf{Y}_{c_\mathbf{k}}\}$, here $\mathbf{Y}_{c_i}$ corresponds to the (*unknown*) label of $\mathbf{X}_{c_i}$ taken from a label set $\mathcal{C} = \{\ell_i\}_i$.

For a given observed superpixel set $\mathbf{X}$, our scene labeling is based on a source-channel model. It defines a joint probability distribution over multiple superpixel *reorderings*, *groupings* (segmentations), *labelings* and finds the best tuple

$$\text{argmax}_{\mathbf{Y},\mathbf{C},\mathbf{k},\pi}\ P(\mathbf{X}, \mathbf{Y}, \mathbf{C}, \mathbf{k}, \pi), \tag{1}$$

where

$$\begin{aligned}
P(\mathbf{X}, \mathbf{Y}, \mathbf{C}, \mathbf{k}, \pi) &= && \qquad (2)\\
P(\pi). & && \text{Superpixel Reordering Model} \qquad (3)\\
P(\mathbf{C}, \mathbf{k}|\pi). & && \text{Superpixel Grouping Model} \qquad (4)\\
P(\mathbf{Y}|\mathbf{C}, \mathbf{k}, \pi). & && \text{Label Dependency Model} \qquad (5)\\
P(\mathbf{X}|\mathbf{Y}, \mathbf{C}, \mathbf{k}, \pi). & && \text{Visual Model.} \qquad (6)
\end{aligned}$$

Here $\pi \in \mathcal{G}(\mathcal{V})$ denotes a permutation (reordering) that maps each element $i \in \mathcal{V}$ to $\pi_i \in \mathcal{V}$ and $\mathcal{G}(\mathcal{V})$ denotes the symmetric group on $\mathcal{V}$ including all the bijections (permutations) from $\mathcal{V}$ to it-self. The model above illustrates a generative scene labeling process which first (i) reorders (*in multiple ways*) the lattice $\mathcal{V}$ resulting into $\pi_1 \dots \pi_n$, (ii) partitions/groups (*in multiple ways*) $\pi_1 \dots \pi_n$ into $\mathbf{k}$ subsets $c_1 \dots c_\mathbf{k}$, then (iii) emits (*in multiple ways*) label hypotheses $\mathbf{Y}_{c_1} \dots \mathbf{Y}_{c_\mathbf{k}}$ for the segments $\mathbf{X}_{c_1} \dots \mathbf{X}_{c_\mathbf{k}}$ and finally (iv) estimates their visual likelihood so only relevant labels will be strengthened. Example shown in Fig. (1, left) illustrates one realization of this stochastic process.

Note that a naive and brute force generation of all the possible reorderings, groupings, labelings would be out of hand; a variant of this process has been achieved in closely related work (see for instance [26]) but turned out to be either tedious or "suboptimal", i.e., none of segments generated are guaranteed to correspond to the actual objects of the scene. As detailed in the subsequent sections, our approach does not explicitly generate all the possible configurations of reordering, grouping and labeling, but instead *implicitly specifies these configurations using finite state machines* in order to build a scored lattice of possible segmentations and labelings.

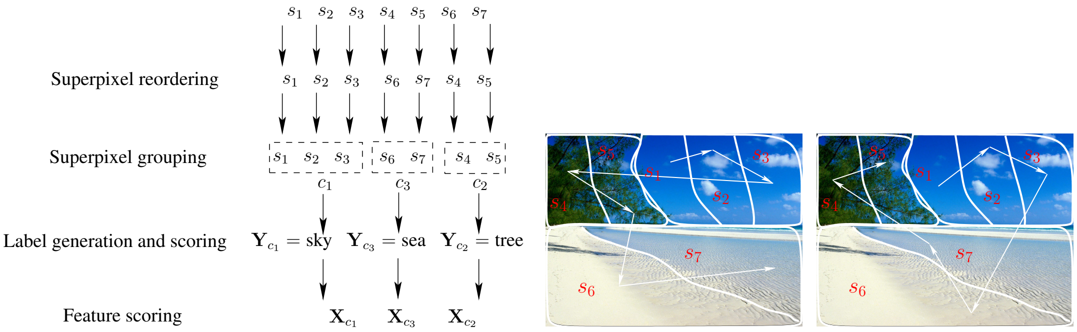

s_1 s_2 s_3 s_4 s_5 s_6 s_7
Superpixel reordering
s_1 s_2 s_3 s_6 s_7 s_4 s_5
Superpixel grouping
s_1 s_2 s_3 c_1
s_6 s_7 c_3
s_4 s_5 c_2
Label generation and scoring
$\mathbf{Y}_{c_1}$ = sky $\mathbf{Y}_{c_3}$ = sea $\mathbf{Y}_{c_2}$ = tree
Feature scoring
$\mathbf{X}_{c_1}$ $\mathbf{X}_{c_3}$ $\mathbf{X}_{c_2}$
s_1 s_2 s_3 s_4 s_5 s_6 s_7
s_1 s_2 s_3 s_4 s_5 s_6 s_7

Figure 1: (Left) this figure shows one realization of the stochastic image labeling process. (Right) this figure shows two parsing possibilities corresponding to two different superpixel permutations.

2.1 Reordering & Grouping Models for Multiple Image Segmentation

Multiple image segmentation is the process of finding multiple partitions of an image lattice $\mathcal{V}$. It is easy to see that the generation of *all the possible* partitions of $\mathcal{V}$ is out of hand; in practice for a lattice including n superpixels, the number of all possible partitions, known as the bell number, is $\frac{1}{e}\sum_{k=0}^{\infty}\frac{k^n}{k!}$ (this number grows fast and reaches $115,975$ partitions with only $n = 10$ superpixels). A majority of these partitions correspond to disconnected segments, with small cardinalities and heterogeneous contents.

In this work, we restrict image partitions to include only connected segments with homogeneous superpixels. More specifically, we generate multiple image segmentations (i) by *reordering* superpixels and (ii) by *grouping* them into connected segments; note that pixels inside superpixels are not reordered. Reordering corresponds to permutations that transform an image lattice $\mathcal{V}$ into many words in $\{\pi\}_{\pi\in\mathcal{G}(\mathcal{V})}$, while grouping breaks every word π into $\mathbf{k}$ subwords. Applying all the permutations $\{\pi\}_{\pi\in\mathcal{G}(\mathcal{V})}$, followed by all the possible grouping makes it possible to generate all the possible partitions of $\mathcal{V}$. Subsets in each partition (again denoted $\mathbf{C} = \{c_i\}_i$) are defined as $c_i = \pi_k^\ell$; here π_k^ℓ refers to a non empty subsequence (or subword) in $\pi_1 \dots \pi_n$ that begins with the k^{th} element and ends with the ℓ^{th} (with $k \leq \ell$), for instance if π corresponds to the sequence $"2341"$, then $\pi_2^3 = "34"$.
These steps (i)+(ii), also shown in Eqs. 3 and 4, are necessary not only to delimit segment boundaries with a high precision but also to evaluate label dependencies between segments at multiple orders (see Section 2.2). Example in Fig. (1, right) illustrates the principle from reordering three segments $c_1 = \{s_1, s_2, s_3\}$, $c_2 = \{s_4, s_5\}$ and $c_3 = \{s_6, s_7\}$. If the underlying labels ($\mathbf{Y}_{c_1}$ = sky, $\mathbf{Y}_{c_2}$ = tree) are less likely to co-occur than ($\mathbf{Y}_{c_1}$ = sky, $\mathbf{Y}_{c_3}$ = sea), then one should reorder them as c_1, c_3, c_2 prior to estimate their dependency statistics using the chain rule; i.e., assuming a first order Markov process, one should consider $P(\mathbf{Y}_{c_1}).P(\mathbf{Y}_{c_3}|\mathbf{Y}_{c_1}).P(\mathbf{Y}_{c_2}|\mathbf{Y}_{c_3})$ instead of $P(\mathbf{Y}_{c_1}).P(\mathbf{Y}_{c_2}|\mathbf{Y}_{c_1}).P(\mathbf{Y}_{c_3}|\mathbf{Y}_{c_2})$. Note also that parsing (2D) images should not be achieved as 1D patterns (such as speech or text) since the order in images is obviously not unique[2].

Reordering & Grouping Models. In order to restrict image partitions to include only connected segments, we consider a random walk generator. First, a given scene is modeled with a graph $G = (V, E)$ where nodes in V correspond to superpixels and edges in E connect superpixels that share common boundaries. Then, our superpixel grouping follows the random walk process: it randomly visits superpixels in G, and groups *only* connected and visually similar ones. Each random walk corresponds to a path in G which is not necessarily Hamiltonian. If one restricts the random walk to include only permutations, then the resulting paths will be Hamiltonian[3] and correspond to partitions of $\mathcal{V}$ that necessarily include connected subsets.

[2] Note that reordering could be related to stochastic models of visual attention and the way the human visual system parses scenes (see for instance [9].)

[3] Note that planar 4-connected graphs (including 4-connected regular grids) are necessarily Hamiltonian.

Considering the lattice $\mathcal{V} = \{1, \dots, n\}$, our reordering & grouping model $P(\mathbf{C}, \mathbf{k}, \pi)$ may be written as

$$P(\mathbf{C}, \mathbf{k}, \pi) = P(\mathbf{C}, \mathbf{k}|\pi) P(\pi), \tag{7}$$

here *our reordering model* $P(\pi)$ equally weights permutations in $\mathcal{G}(\mathcal{V})$, i.e., $\forall \pi \in \mathcal{G}(\mathcal{V})$, $P(\pi) = 1/|\mathcal{G}(\mathcal{V})|$. Considering *our grouping model* $P(\mathbf{C}, \mathbf{k}|\pi)$ as a first order Markov process, and assuming all segments in a given partition conditionally independent given π, one may write

$$P(\mathbf{C}, \mathbf{k}|\pi) = P(\mathbf{k}) \prod_{i=1}^{\mathbf{k}} P(c_i|\pi). \tag{8}$$

All the partition sizes have the same mass, i.e., $P(\mathbf{k}) = 1/n, \forall\ \mathbf{k} \in \{1, 2, \dots, n\}$ and

$$P(c_i|\pi) \quad = \quad P(\pi_{k_i}) \prod_{j=1}^{\ell_i - k_i} P(\pi_{k_i+j}|\pi_{k_i+j-1}). \tag{9}$$

Each partition $\mathbf{C}$ is defined a $\{c_i = \pi_{k_i}^{\ell_i}\}_{i=1}^{\mathbf{k}}$ with $\big\{(k_i, \ell_i)\big\}_{i=1}^{\mathbf{k}}$ satisfying $k_1 = 1$, $\ell_{\mathbf{k}} = n$, $k_i = \ell_{i-1} + 1, \forall i \in \{2, \dots, \mathbf{k}\}$ and $k_i \leq \ell_i$. In the above probability, $P(\pi_{k_i})$ is taken as uniform, i.e., $1/n$ and $P(\pi_{k_i+j}|\pi_{k_i+j-1})$ is taken as the random walk transition probability from node (superpixel) π_{k_i+j-1} to node π_{k_i+j}, which is positive only if the underlying superpixels share a common boundary; and it is set to $P(\pi_{k_i+j}|\pi_{k_i+j-1}) \ \propto\ \mathbb{1}_{\{(\pi_{k_i+j}, \pi_{k_i+j-1}) \in E\}} \cdot \kappa\big(\psi(\pi_{k_i+j}), \psi(\pi_{k_i+j-1})\big)$; here κ is the histogram intersection kernel and $\psi(\pi_{k_i+j})$ denotes a visual feature extracted at superpixel π_{k_i+j}. Again, this conditional probability of transition between superpixels π_{k_i+j}, π_{k_i+j-1} depends on whether they are neighbors in G and also on their visually similarity. Put differently, if the transition between neighboring superpixels is achieved with a conditional probability larger than uniform, then these superpixels are considered as visually similar and *likely* to come from the same physical segment (object) in the scene.

2.2 Visual and Label Dependency Models

Once superpixels reordered and grouped in multiple ways, we use a unary visual model and label interaction potentials, described below, in order to score the resulting partitions. As shown in the remainder of this paper, only highly scored partitions are likely to correspond to correct object segmentations.

Label Dependency Model. this model captures scene structure and a priori knowledge about segment/label relationships (either co-occurrence or geometric relationships) in order to consolidate labels which are consistent with already observed scenes.

Considering a first order Markovian process and using the chain rule, our bi-gram label dependency model is

$$P(\mathbf{Y}|\mathbf{C}, \mathbf{k}, \pi) \quad = \quad P(\mathbf{Y}_{c_1}) \prod_{i=2}^{\mathbf{k}} P(\mathbf{Y}_{c_i}|\mathbf{Y}_{c_{i-1}}).$$

Given two disjoint segments $c_i = \{\pi_{k_i}, \dots, \pi_{\ell_i}\}$, $c_j = \{\pi_{k_j}, \dots, \pi_{\ell_j}\}$ as a union of superpixels with arbitrary orders, we have

$$\begin{array}{rcl} P(\mathbf{Y}_{c_i}) & = & P(\mathbf{Y}_{\pi_{k_i}} \dots \mathbf{Y}_{\pi_{\ell_i}}) \\ & := & P(\mathbf{Y}_{\pi_{k_i}}) \displaystyle\prod_{q=1}^{\ell_i - k_i} P(\mathbf{Y}_{\pi_{k_i+q}}|\mathbf{Y}_{\pi_{k_i+q-1}}) \\ P(\mathbf{Y}_{c_i}|\mathbf{Y}_{c_j}) & = & P(\mathbf{Y}_{\pi_{k_i}} \dots \mathbf{Y}_{\pi_{\ell_i}}|\mathbf{Y}_{\pi_{k_j}} \dots \mathbf{Y}_{\pi_{\ell_j}}) \\ & := & \left(P(\mathbf{Y}_{\pi_{k_i}}) \displaystyle\prod_{q=1}^{\ell_i - k_i} P(\mathbf{Y}_{\pi_{k_i+q}}|\mathbf{Y}_{\pi_{k_i+q-1}})\right) \\ & & P(\mathbf{Y}_{\pi_{k_j}}|\mathbf{Y}_{\pi_{\ell_i}}) \left(\displaystyle\prod_{q=1}^{\ell_j - k_j} P(\mathbf{Y}_{\pi_{k_j+q}}|\mathbf{Y}_{\pi_{k_j+q-1}})\right). \end{array} \tag{10}$$

Let $\mathcal{I} = \{\mathbf{I}_1, \dots, \mathbf{I}_N\}$ be a training set, of fixed size images, labeled at the pixel level (with labels in $\mathcal{C}$). Let $\mathbf{f}_u(\ell, p) = \sum_{i=1}^{N} \mathbb{1}_{\{\mathbf{I}_i(p) = \ell\}}$ be the frequency of co-occurrence of pixel p and label ℓ in

$\mathcal{I}$. Similarly, we define $\mathbf{f}_b(\ell, \ell', p, p')$ as $\sum_{i=1}^{N} \mathbb{1}_{\{\mathbf{I}_i(p)=\ell\}} \mathbb{1}_{\{\mathbf{I}_i(p')=\ell'\}}$. Let's denote the labels of two given superpixels s, s' respectively as $\mathbf{Y}_s$, $\mathbf{Y}_{s'}$, we define

$$\begin{aligned} P(\mathbf{Y}_s) &= \frac{\sum_{p \in s} \mathbf{f}_u(\mathbf{Y}_s, p)}{\sum_{\ell \in \mathcal{C}} \sum_{p \in s} \mathbf{f}_u(\ell, p)} \\ P(\mathbf{Y}_s | \mathbf{Y}_{s'}) &= \frac{\sum_{p \in s} \sum_{p' \in s'} \mathbf{f}_b(\mathbf{Y}_s, \mathbf{Y}_{s'}, p, p')}{\sum_{\ell \in \mathcal{C}} \sum_{p \in s} \sum_{p' \in s'} \mathbf{f}_b(\ell, \mathbf{Y}_{s'}, p, p')}. \end{aligned} \tag{11}$$

Visual Model. this model defines the likelihood of $\mathbf{X} = \{\mathbf{X}_{\mathbf{c_1}}, \dots, \mathbf{X}_{\mathbf{c_k}}\}$ given the labels $\mathbf{Y} = \{\mathbf{Y}_{\mathbf{c_1}}, \dots, \mathbf{Y}_{\mathbf{c_k}}\}$. Assuming conditionally independent superpixels and segments given their labels, and assuming that each $\mathbf{X}_{c_i}$ depends only on $\mathbf{Y}_{c_i}$, we define our visual model as

$$P(\mathbf{X}|\mathbf{Y}, \mathbf{C}, \mathbf{k}, \pi) = \prod_{i=1}^{\mathbf{k}} \prod_{s \in c_i} P(\mathbf{X}_s | \mathbf{Y}_{c_i}) \tag{12}$$

$$\text{with} \quad P(\mathbf{X}_s | \mathbf{Y}_{c_i}) \propto \frac{1}{1 + \exp(-f_{\mathbf{Y}_{c_i}}(\mathbf{X}_s))}, \tag{13}$$

here $f_{\mathbf{Y}_{c_i}}(.)$ is an SVM classifier (based on histogram intersection kernel) trained to discriminate superpixels belonging to a category $\mathbf{Y}_{c_i}$ from $\mathcal{C} \backslash \mathbf{Y}_{c_i}$. In practice, each superpixel $\mathbf{X}_s$ is described using the bag-of-word SIFT representation. Precisely, SIFT features are extracted and quantized using a codebook of 200 visual words and a two level spatial pyramid is extracted on each superpixel resulting into a feature vector of 1,000 dimensions.

3 Finite State Machine Inference

In this section, we implement the models discussed earlier using finite state machines. We first remind the definition of stochastic finite state machines, then we show how we design and combine those machines in order to build a global transducer. The latter encodes in a compact way, the lattice of possible annotations of a given scene and the best one corresponds to the best path in that lattice.

3.1 Probabilistic Finite State Machines

Some of the variables used for notation (mainly π, p, q, etc.) are reused and will not be confused with the ones used earlier. In this section, we define two particular FSMs: finite state transducters (FSTs) and finite state acceptors (FSAs).

A probabilistic finite state transducer is a tuple $A = (Q, \Sigma_I, \Sigma_O, \delta, I, F)$, where Q is a finite set of states, $I \subseteq Q$ is the set of initial states, $F \subseteq Q$ is the set of final states, and Σ_I, Σ_O are two alphabets (not necessarily equal). δ is a finite set of transitions of the form $q \xrightarrow{a:b/p} q'$ where q and q' are states in Q, $a \in \Sigma_I$ and $b \in \Sigma_O$ are two letters (a, b may also be the empty word ϵ) and p is a probability; if the set of outgoing transitions from q is $\{q \xrightarrow{a_1:b_1/p_1} q_1, \dots, q \xrightarrow{a_m:b_m/p_m} q_m\}$, then $\sum\limits_{i=1}^{m} p_i = 1$.

A path π of A is a sequence of transitions of the form

$$q_1 \xrightarrow{a_1:b_1/p_1} q_2 \xrightarrow{a_2:b_2/p_2} q_3 \cdots q_{m-1} \xrightarrow{a_{m-1}:b_{m-1}/p_{m-1}} q_m$$

labeled by $(a_1, b_1) \cdots (a_{m-1}, b_{m-1})$ with probability $\prod_{i=1}^{m-1} p_i$.

π is accepting if $q_1 \in I$ is an initial state and $q_m \in F$ is a final state. A word $w \in (\Sigma_I \times \Sigma_O)^*$ is accepted by A with probability p iff there exists an integer k such that A has exactly k transitions labeled by w, with probabilities $p_1, \dots, p_k$, respectively and $p = \prod_{i=1}^{k} p_i$.

A probabilistic automaton A defines a distribution μ as follows: if w is accepted by A with probability p, then $\mu(w) = p$. If w is not accepted by A, then $\mu(w) = 0$. The language of A (denoted $\mathcal{L}(A)$) is the set of all the accepted words of A.

When all paths of A are unweighted and of the form $q_1 \xrightarrow{a_1} q_2 \xrightarrow{a_2} q_3 \cdots q_{m-1} \xrightarrow{a_{m-1}} q_m$, A will be referred to as finite state acceptor and it is labeled by the word $a_1 \cdots a_{m-1}$; the alphabet of A is denoted simply as Σ.

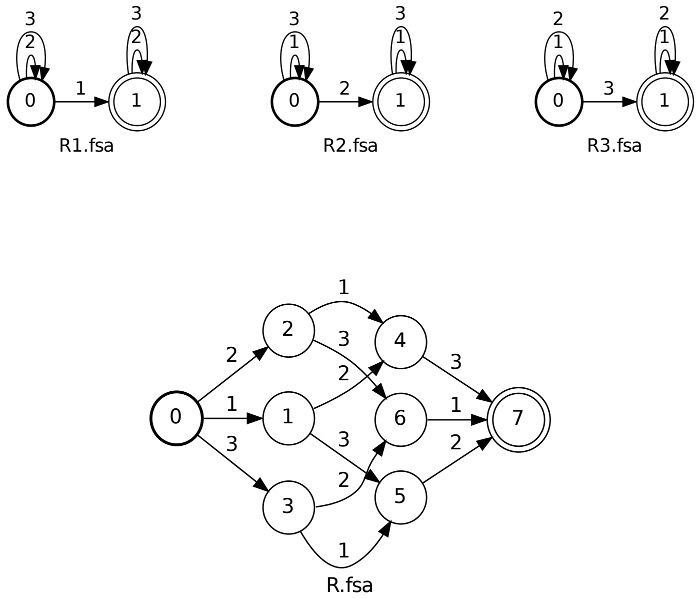


Figure 2: This figure shows elementary FSAs combined (via intersection) in order to build the reordering FSA $\mathbf{R}$.

3.2 "Reordering" FSA

The "reordering" FSA (denoted $\mathbf{R}$) makes it possible to generate superpixel permutations prior to group them in order to generate multiple image segmentations. Let $\mathbf{R}_i$ denote an FSA which generates all the possible superpixel sequences containing only one instance of a superpixel i at any position in $\mathbb{N}^+$, the language of $\mathbf{R}_i$ is

$$\mathcal{L}(\mathbf{R}_i) = \Big(\bigcup_{j \neq i} j\Big)^* \,.\, i \,.\, \Big(\bigcup_{j \neq i} j\Big)^*,$$

here "." stands for superpixel concatenation and $()^*$ stands for zero or multiple concatenations of superpixels. The alphabet set of $\mathbf{R}_i$ is defined as $\Sigma = \{1, \ldots, n\}$, the initial and final states as $I = \{q_0\}$, $F = \{q_1\}$, $Q = \{q_0, q_1\}$, and the underlying set of transitions is

$$\delta_i = \Big\{ q_0 \xrightarrow{j} q_0 \Big\}_{j \neq i} \bigcup \Big\{ q_0 \xrightarrow{i} q_1 \Big\} \bigcup \Big\{ q_1 \xrightarrow{j} q_1 \Big\}_{j \neq i}.$$

Now, the "reordering" finite state machine $\mathbf{R}$ is obtained via intersections[4] as

$$\mathbf{R} = \mathbf{R}_1 \cap \cdots \cap \mathbf{R}_n$$

The graphical representation of the FSAs $\mathbf{R}_1$, $\mathbf{R}_2$, $\mathbf{R}_3$ (for $n = 3$) as well as their intersection $\mathbf{R}$, is depicted in Fig. 2. The FSA $\mathbf{R}$ makes it possible to generate all the reorderings each one includes n superpixels.

3.3 "Superpixel Grouping" FST

Given a scene $\mathbf{I}$, we define a graph (V, E) where each node $v_i \in V$ corresponds to a superpixel i and an edge $e_{ij} \in E$ connects two superpixels i, j if they share a common boundary in $\mathbf{I}$. The

[4] Details about elementary algebra for automata including "intersection, union and composition" are of course out of the scope of this work. Comprehensive reviews of these issues can be found, for instance, in [8].

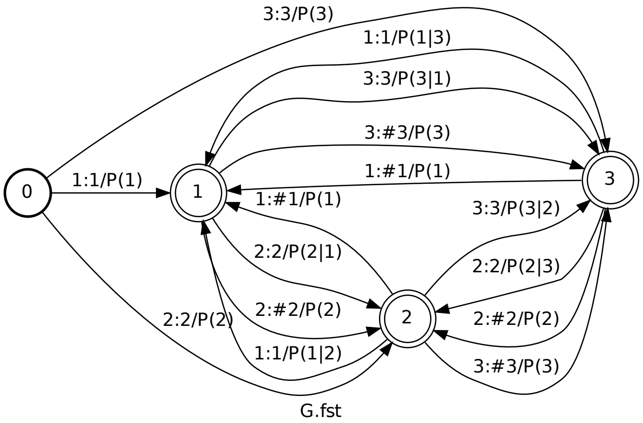


Figure 3: This figure shows the grouping FST **G**.

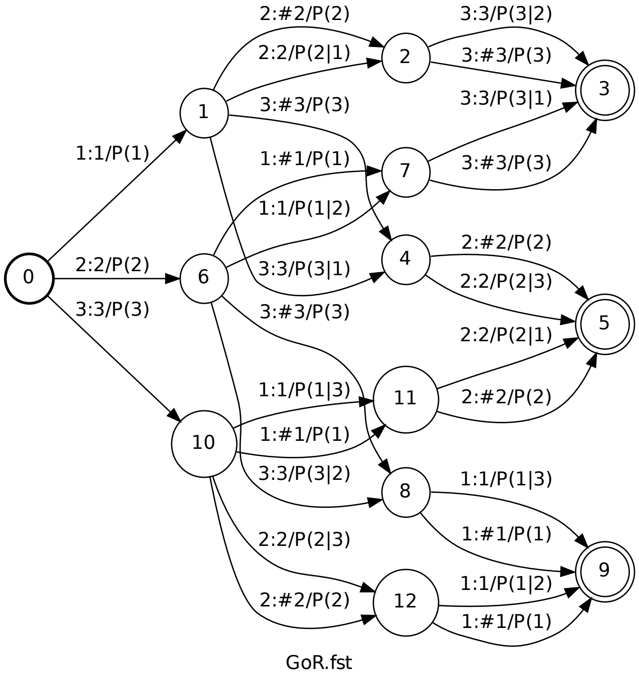


Figure 4: This figure shows composition of the reordering FSA (**R**) and the grouping FST (**G**).

"grouping" FST (denoted $\mathbf{G}$) generates random walks in (V, E) and enumerates the underlying superpixels; the set of all possible superpixel enumerations is denoted $\mathcal{L}(\mathbf{G})$. When $\mathbf{G}$ is applied to the outputs of the "reordering" $\mathbf{R}$ (see section 3.2), the resulting enumerations are referred to as *valid superpixel enumerations*.

Multiple partitions of $\mathbf{I}$ are generated by splitting *in multiple ways* each valid superpixel enumeration in $\mathcal{L}(\mathbf{G})$. In practice, splitting is achieved by adding separators (denoted $\#$) between superpixels at random locations (in $\{1, \dots, n\}$) so superpixels delimited by two successive separators will belong to the same segment. Note that when the graph (V, E) is not complete, the partitions generated according to this process do not necessarily correspond to all the possible segmentations of $\mathbf{I}$ as only a subset of them is kept in accordance with the random walk generator, i.e., only partitions including connected segments are allowed. This restriction about the connectivity of segments is very reasonable as real world objects are assumed connected, even though exceptions may be found; furthermore the growth of the number of possible partitions of $\mathbf{I}$ with respect to n is much slower than the total number of possible partitions of $\mathbf{I}$ (i.e., including those which are not connected) and this makes the process of generating multiple partitions more efficient.

The definition of the "grouping" FST $\mathbf{G}$ is given as $\Sigma_I = \{1, \dots, n\}$ and $\Sigma_O = \Sigma_I \cup \{\#\}$. Transitions of this FST are defined as

$$\delta = \underbrace{\left\{ q_0 \xrightarrow{i:i/P(i)} q_i,\ q_i \xrightarrow{j:j/P(j|i)} q_j \right\}_{i \sim j}}_{\text{Grouping actions}} \bigcup \underbrace{\left\{ q_i \xrightarrow{j:\#j/P(j)} q_j \right\}_{i \sim j}}_{\text{Splitting actions}},$$

here $i \sim j$ means that superpixels i, j share a common boundary in $\mathbf{I}$. The set of initial and final states are defined as $I = \{q_0\}$, $F = \{q_i\}_i$ and $Q = \{q_0\} \cup \{q_i\}_i$. This FST may either i) group two superpixels j, i with a probability $P(j|i)$ of a walk from i to j or ii) start a new sequence of superpixels at j (separate j from i) with a probability $P(j)$ (splitting action). The graphical representation of this FST is depicted in example of Fig. 3. The FST resulting from the composition $\mathbf{G} \circ \mathbf{R}$ (see Fig. 4) makes it possible to achieve multiple grouping of superpixels taken at different orders, so multiple partitions (of continuous segments) will be generated.

3.4 "Label Dependency" FST

The label dependency FST is designed in order to provide us with the probability of a sequence of labels given a partition of superpixels. Using the chain rule as described earlier in section 2.2, this may be written using unary and dependency statistics. The implementation of this dependency FST (denoted $\mathbf{D}$), is achieved by reading the outputs (partitions) of the FST ($\mathbf{G} \circ \mathbf{R}$) and emitting for each partition of superpixels a sequence of labels, resulting into the following transitions

$$\delta = \underbrace{\left\{ q_i \xrightarrow{j:jc_i} q_i \right\}_i}_{\text{Read \& assign superpixel } j \text{ to } c_i} \bigcup \underbrace{\left\{ q_0 \xrightarrow{\#:\mathbf{Y}_{c_1}/P(\mathbf{Y}_{c_1})} q_1 \right\}}_{\text{Return labels with unary statistics}} \bigcup \underbrace{\left\{ q_{i-1} \xrightarrow{\#:\mathbf{Y}_{c_i}/P(\mathbf{Y}_{c_i}|\mathbf{Y}_{c_{i-1}})} q_i \right\}_i}_{\text{Return labels with dependency statistics}}.$$

Again $\#$ corresponds to segment end. The alphabet sets of $\mathbf{G}$ are $\Sigma_I = \{1, \dots, n\} \cup \{\#\}$, $\Sigma_O = \{\ell_i\}_i \cup \{jc_i\}_{i,j}$ (with jc_i stands for superpixel j belongs to segment c_i) and $I = \{q_0\}$, $F = \{q_i\}_{i \neq 0}$, $Q = \{q_i\}_i$ while $P(\mathbf{Y}_{c_1})$, $P(\mathbf{Y}_{c_i}|\mathbf{Y}_{c_{i-1}})$ are described in section 2.2. Figure 5 shows an example of this FST.

3.5 "Visual" FST

Different FSTs are introduced in the previous sections allowing to generate multiple segmentations and to label each of them. Among these segmentations and labeling, only few of them are "likely to occur" according to the visual model. As described in section 2.2, for each label $\mathbf{Y}_{c_i} \in \{\ell_i\}_i$, we train a visual model, that provides for every superpixel $j \in c_i$, its likelihood conditioned by $\mathbf{Y}_{c_i}$. The implementation of this visual FST (denoted $\mathbf{V}$) is achieved by parsing all the superpixels $j \in c_i$ generated by the FST ($\mathbf{D} \circ \mathbf{G} \circ \mathbf{R}$), and returning their labels and visual likelihoods $P(\mathbf{X}_j|\mathbf{Y}_{c_i})$. This results into the following rules

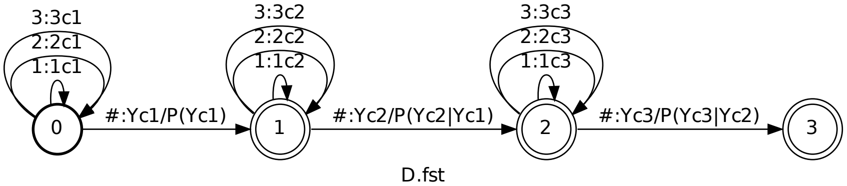


Figure 5: This figure shows the dependency FST ($\mathbf{D}$); 3c1 stands for 3^{rd} superpixel belongs to segment c_1.

$$\delta = \underbrace{\left\{ q_0 \xrightarrow{jc_i:\epsilon/P(\mathbf{X}_j|\mathbf{Y}_{c_i})} q_0 \right\}_{i,j}}_{\text{Read \& score the } j^{th} \text{ superpixel in } c_i} \bigcup \underbrace{\left\{ q_0 \xrightarrow{\mathbf{Y}_{c_i}:\mathbf{Y}_{c_i}} q_0 \right\}_{i}}_{\text{Return labels}} .$$

In this FST $\mathbf{V}$, $\Sigma_I = \{\ell_i\}_i \cup \{jc_i\}_{i,j}$, $\Sigma_O = \{\ell_i\}_i$, and $Q = I = F$ correspond to the singleton $\{q_0\}$. Figure 6 shows an example of this FST.

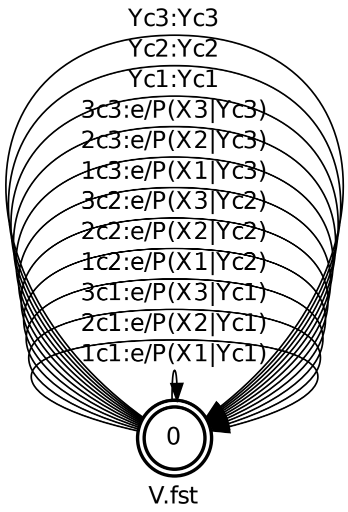


Figure 6: This figure shows the visual model FST ($\mathbf{V}$). Again, 3c1 stands for 3^{rd} superpixel belongs to segment c_1 (the empty word ϵ is also denoted as e).

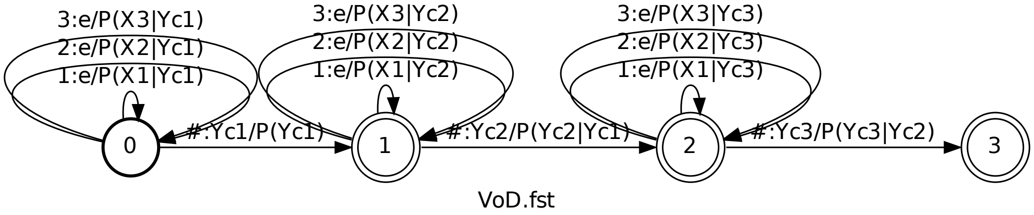


Figure 7: This figure shows the composition of the dependency FST ($\mathbf{D}$) and the visual model FST ($\mathbf{V}$).

3.6 The "Global" FSM

Given a scene paved with a collection of non-overlapping superpixels, our labeling model first reorders these superpixels (via the "reordering FSA" $\mathbf{R}$) and group them (via the "grouping FST" $\mathbf{G}$). These two FSMs when composed together, allow us to generate many reordered partitions of segments[5]. Among these partitions, only a few of them are relevant and correspond to *meaningful* objects in the scene. Therefore, and in order to find these relevant partitions, we combine the $\mathbf{R}$ and $\mathbf{G}$ FSMs with another FSM (resulting from $\mathbf{V} \circ \mathbf{D}$; see example in Fig. 7) that scores segments in all possible partitions, depending on their *unary* and *high order* interactions, and returns only the most likely partition and its labels. The most likely solution (partition and its labels) corresponds to the best (shortest) path in the global FSM ($\mathbf{V} \circ \mathbf{D} \circ \mathbf{G} \circ \mathbf{R}$) provided that negative log-likelihood transform is applied to all FSM transition probabilities; this solution also minimizes the energy function $-\log P(\mathbf{X}, \mathbf{Y}, \mathbf{C}, \mathbf{k}, \pi)$ (see Eq. 1). Figure 8 shows an example of this global FSM.

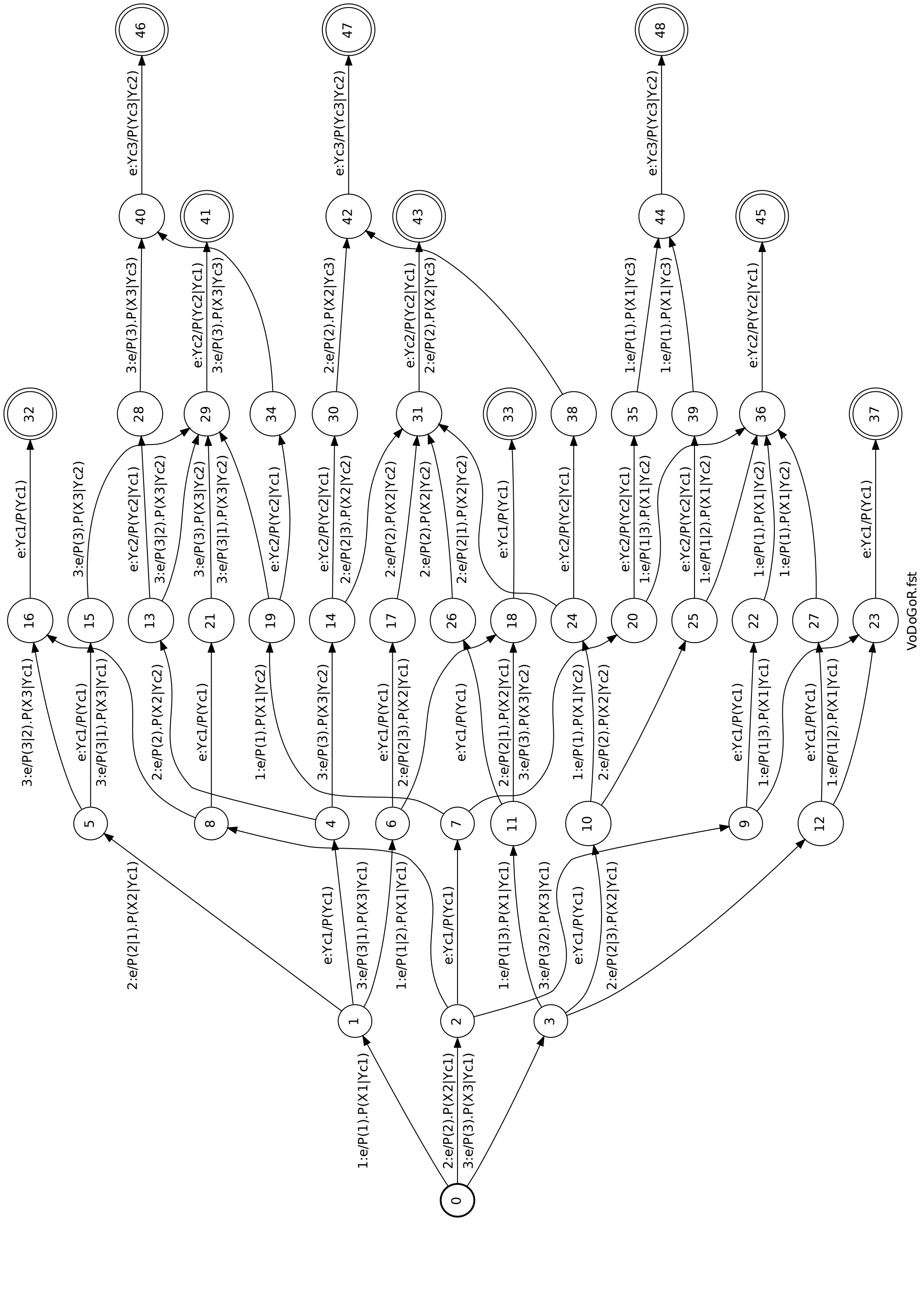
[5] Each segment in these partitions is seen as a "phrase" in a "language". The alphabet of this language corresponds to all the superpixels of that scene.

Again, the global FSM (denoted $\mathbf{F}$) results from the composition of all the previous FSMs ($\mathbf{F} = \mathbf{V} \circ \mathbf{D} \circ \mathbf{G} \circ \mathbf{R}$) and this order of application is strict, i.e., first reordering of superpixels is generated, then superpixel grouping is achieved in order to generate multiple segmentations, followed by multiple label generation for every partition, and finally only a few of these labels are kept according to both the label dependency and the visual models. Notice that this process may end-up with a complex FSM $\mathbf{F}$ (i.e., the composition process is time and memory demanding). One may reduce the complexity of different transducers (and thereby the global one) at different levels including the visual and the label dependency models by only keeping sparse transitions (related to strictly positive or large statistics). Another possible simplification consists in reducing the number of possible labels, especially if one is interested in domain specific applications with restricted labels.

4 Conclusion

In this paper, we introduced a complete framework for scene parsing and annotation based on finite state machines. The approach is complete and allows us to examine and score more “exhaustive” configurations in order to achieve segmentation as well as annotation.
Future work includes the application of the proposed method using real-world and challenging benchmarks.

Figure 8: This figure shows the global finite state machine $\mathbf{V} \circ \mathbf{D} \circ \mathbf{G} \circ \mathbf{R}$.



References

[1] K. Barnard, P.Duygululu, D. Forsyth, D. Blei, and M. Jordan. Matching words and pictures. *The Journal of Machine Learning Research*, 2003.

[2] D. Batra, R. Sukthankar, and C. Tsuhan. Learning class-specific affinities for image labelling. *in Proc. CVPR*, 2008.

[3] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *In IEEE Transactions on PAMI*, 26(9):1124–1137, sep 2004.

[4] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *In IEEE transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[5] G. Carneiro and N. Vasconcelos. Formulating semantic image annotation as a supervised learning problem. *in Proc. of CVPR*, 2005.

[6] L.-B. Chang, Y. Jin, W. Zhang, E. Borenstein, and S. Geman. Context, computation, and optimal roc performance in hierarchical models. *International journal of computer vision*, 93(2):117–140, 2011.

[7] P. Duygulu, K. Barnard, J. deFreitas, and D. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. *In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2353, pp. 97-112. Springer, Heidelberg*, 2002.

[8] S. Eilenberg. *Automata, languages, and machines*, volume 1. Academic press, 1974.

[9] W. Einhaeuser, T. N. Mundhenk, P. F. Baldi, C. Koch, and L. Itti. A bottom-up model of spatial attention predicts human error patterns in rapid scene recognition. *Journal of Vision*, 7(10):1–13, 2007.

[10] A. Farag, A. El-Baz, and G. Gimel'farb. Precise segmentation of multi-modal images. *IEEE Trans. on Image Processing*, 15(4):952–968, 2006.

[11] S. Feng, R. Manmatha, and V. Lavrenko. Multiple bernoulli relevance models for image and video annotation. *In: Proc. of ICCV, pp. 1002-1009*, 2004.

[12] C. Galleguillos, A. Rabinovich, and S. Belongie. Object categorization using co-occurrence, location and appearance. *in Proc. CVPR*, 2008.

[13] Y. Gao, J. Fan, X. Xue, and R. Jain. Automatic image annotation by incorporating feature hierarchy and boosting to scale up svm classifiers. *in Proc. of ACM MULTIMEDIA*, 2006.

[14] X. He, R. Zimel, and M. Carreira. Multiscale conditional random fields for image labeling. *In CVPR*, 2004.

[15] J. Jeon, V. Lavrenko, and R.Manmatha. Automatic image annotation and retrieval using cross-media relevance models. *in Proc. of ACM SIGIR*, pages 119–126, 2003.

[16] P. Kohli, L. Ladicky, and P. Torr. Robust higher order potentials for enforcing label consistency. *in Proc. CVPR*, 2008.

[17] L. Ladicky, C. Russell, and P. Kohli. Associative hierarchical crfs for object class image segmentation. *in Proc ICCV*, 2009.

[18] V. Lavrenko, R. Manmatha, and J. Jeon. A model for learning the semantics of pictures. *In: Proc. of NIPS*, 2004.

[19] J. Li and J. Z. Wang. Automatic linguistic indexing of pictures by a statistical modeling approach. *IEEE Trans. on PAMI*, 25(9):1075–1088, 2003.

[20] X. Li and H. Sahbi. Superpixel based object class segmentation using conditional random fields. *In the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2011.

[21] J. Liu, M. Li, Q. Liu, H. Lu, and S. Ma. Image annotation via graph learning. *Pattern Recognition*, 42(2):218–228, 2009.

[22] J. Liu, B. Wang, M. Li, Z. Li, W. Ma, H. Lu, and S. Ma. Dual cross-media relevance model for image annotation. *In Proc. of ACM MULTIMEDIA, pp. 605-614*, 2007.

[23] F. Monay and D. GaticaPerez. Plsa-based image autoannotation: Constraining the latent space. *in Proc. of ACM International Conference on Multimedia*, 2004.

[24] G. Moser and B. Serpico. Combining support vector machines and markov random fields in an integrated framework for contextual image classification. *In TGRS*, 2012.

[25] S. Nowak and M. Huiskes. New strategies for image annotation: Overview of the photo annotation task at imageclef 2010. *in The Working Notes of CLEF 2010*, 2010.

[26] C. Pantofaru, C. Schmid, and M. Hebert. Object recognition by integrating multiple image segmentations. *in Proc. ECCV*, 2008.

[27] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. *in Proc. ICCV*, 2007.

[28] J. Reynolds and K. Murphy. Figure-ground segmentation using a hierarchical conditional random field. *in Proc. Fourth Canadian Conference on Computer and Robot Vision*, 2007.

[29] H. Sahbi and X. Li. Context based support vector machines for interconnected image annotation (the saburo tsuji best regular paper award). *In the Asian Conference on Computer Vision (ACCV)*, 2010.

[30] D. Semenovich and A. Sowmya. Geometry aware local kernels for object recognition. *In ACCV*, 2010.

[31] V. D. Shet, J. Neumann, V. Ramesh, and L. S. Davis. Bilattice-based logical reasoning for human detection. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[32] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. *in Proc. CVPR*, 2008.

[33] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. *in Proc. ECCV*, pages 1–15, 2006.

[34] A. Singhal, L. Jiebo, and Z. Weiyu. Probabilistic spatial context models for scene content understanding. *In CVPR*, 2003.

[35] S. Todorovic and N. Ahuja. Learning subcategory relevances for category recognition. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[36] C. Wang, S. Yan, L. Zhang, and H. Zhang. Multi-label sparse coding for automatic image annotation. *in Proc. of CVPR*, 2009.

[37] T. Wu and S.-C. Zhu. A numerical study of the bottom-up and top-down inference processes in and-or graphs. *International journal of computer vision*, 93(2):226–252, 2011.

[38] L. Yang, P. Meer, and D. Foran. Multiple class segmentation using a unified framework over mean-shift patches. *in Proc CVPR*, 2007.

[39] S. C. Zhu and D. Mumford. *A stochastic grammar of images*, volume 2. Now Pub, 2007.

Dépôt légal : 2013 - 2^{e} trimestre
Imprimé à Télécom ParisTech – Paris
ISSN 0751-1345 ENST D (Paris) (France 1983-9999)

Télécom ParisTech

Institut Mines-Télécom - membre de ParisTech

46, rue Barrault - 75634 Paris Cedex 13 - Tél. + 33 (0)1 45 81 77 77 - www.telecom-paristech.fr

Département TSI

© Institut Mines-Télécom -Télécom ParisTech 2013