# A Novel Use of Higher Order Information to Detect Inliers of an Implicit Model

Toufiq Parag and Ahmed Elgammal

Dept of Computer Science, Rutgers University, Piscataway, NJ 08854.
*Email:* {*tparag, elgammal*}*@cs.rutgers.edu*

**Abstract.** The problem we address in this paper is to label datapoints when the information about them is provided primarily in terms of their subsets or groups. The knowledge we have for a group is a likelihood value for each group member to belong to same class. These likelihood values are computed using a model, either explicit or implicit, of the pattern we wish to learn. By defining a Markov Random Field (MRF) over the labels of data, we formulate the problem as an MRF inference problem. A submodular function is defined as clique potential function for this MRF. We identified a special form of this clique potential function that results in a simple and efficient inference method with linear running time with respect to the number of datapoints and subsets. The formulation allows augmentation of energy functions for data with several different size subsets. We present experimental results in applications where the proposed method produces improved performances over other methods.

## 1 Introduction

The objective of this paper is to label datapoints into two classes when the information about the datapoints is available in terms of small groups[1] of data. The information encode the likelihood that all the members of the corresponding group should fall into the same class. These likelihood measures are generated utilizing a model, either analytical or implicit, of the entity or pattern we are trying to decide about. Given the subsets and their corresponding likelihood measures, we wish to find the labels of data having maximal agreement with these measures.

It should be noted here that we are interested in scenarios where higher order knowledge about data subsets of sizes $k > 2$ is the primary source of information for labeling; we may or may not have any information about the individual data sample or pairs of samples. Furthermore, we may have data subsets of different sizes, i.e., $k = 3, 4, \ldots$, each with corresponding likelihood measure, to decide the individual label from.

There is a direct connection between the problem we are dealing with and hypergraph node labeling. Hypergraphs are generalization of graphs where each

[1] Group simply implies a collection of datapoints. The mathematical definition of group is not used in this paper. Similarly, likelihood simply implies a numerical weight.

hyperedge connects more than two vertices, i.e., a hyperedge is a subset of of nodes. Weighted hypergraphs, a hypergraph whose hyperedges are associated with real valued weights, have been recently gained much popularity in computer vision for the purpose of representing the object geometry. Each small group of data, as discussed above, can be considered as a hyperedge of some hypergraph. The likelihood measure for this group of data is analogous to the weight of corresponding hyperedge. Given such a hypergraph, we would like to label its nodes into two categories such that optimal number of nodes, connected by hyperedge with large weights, tend to fall into the same category.

There are many examples of such problem in machine learning and vision literature. In part-based object recognition for computer vision, we may learn an implicit statistical model among groups of detected parts to decide which of the detected parts actually belong to the object we are trying to recognize. It is well known that larger groups of parts capture more geometrical information than pairs of parts. This is a necessary step in recognition due to the fact that object part-detectors often generate many false alarms. In website classification problems, a subset comprising more than two words could be more informative about the category than that comprising one or two words.

Subset-wise information is also utilized in model estimation algorithms. The problem of model estimation is to determine the parameters of a model given a set of (noisy) data and an analytic form of a model. A standard algorithm, namely RANSAC [1], for this sort of problem randomly samples smaller subsets of data, computes target model parameters and calculates an error value for each of the subsets. RANSAC is used for fundamental matrix computation, affine motion estimation etc in computer vision [2] literature. We develop an algorithm that can solve any model estimation problem that RANSAC can handle.

We propose a novel approach to solve this labeling problem in a principled fashion. Each datapoint is associated with a binary label variable. These label variables are modeled as Markov Random Fields (MRF) [3, 4] with appropriate neighborhood system and clique definitions (to be explained later). We propose a (family of) potential function for each clique which increases with the number of mismatches among the labels of clique members.

The choice of our potential function plays the central role in deciding labels of data samples. The potential function increases the penalty of having different labels in the same subset in proportion to the likelihood measure. We prove that this family potential function is submodular and therefore can be minimized in strongly polynomial time [5]. Moreover, we show that for a specific form of this potential function, we can solve the inference problem with a simple algorithm that runs in linear time w.r.t. the number of data samples and subsets. We refer to a linear programming solution for inference problem for other forms of this potential function. This MRF formulation is able to cope with the situation when the likelihood values are available for different size subsets.

In light of the discussion above, the main *contributions* of this paper are (1) to formulate the problem of labeling data primarily from subset-wise information as an MRF inference problem, (2) to propose a submodular clique potential

function so that the inference can be calculated in strongly polynomial time. We also identify a special form of potential function for which we already have a simple and fast inference algorithm. However, this should be clear at this point that we are *NOT* proposing a new optimization method for MRF inference. We are presenting a new view to how MRF can be applied in labeling problems with higher order information and proposing effective clique function for that purpose. Our findings shed a new light on how to extract useful and discriminative information from higher order knowledge. We also show how a special form of the proposed framework can result in a simple algorithm for problems like model estimation.

**Relevant work:** Before describing the method in detail, we would like to contrast our method to hypergraph clustering methods (not described here due to space limitation, see e.g., [6–10]). Intuitively, the proposed work differs from hypergraph clustering in the definition of hyperedge weights. We use the likelihood values calculated given a model as hyperedge weights. On the other hand, clustering problems computes the similarity using distances among datapoints in an unsupervised fashion. Theoretically, the most significant difference between hypergraph clustering and the proposed method is the potential function over each clique. As already stated, this function is primarily responsible for determining the labels. The choice of our clique function offers more flexibility over how we wish to partition the clique itself and the dataset overall. We can also learn and/or adjust the parameters of this function to estimate a specific pattern rather than having 'balanced' clusters. It is possible to use subsets of different cardinalities in the proposed labeling algorithm, which is not addressed in most hypergraph clustering papers.

There are some recent works on higher order MRF inference problems in the literature [11–13]. All these papers examine how can inference for higher order MRFs can be solved by the graph-cut algorithm [14]. Though these studies motivate the design of our potential clique functions, there is a fundamental difference between our work and those of of [11–13]. *We aim to solve a set of problems different from those of [11–13] and we do not use graph-cut for inference.* The difference is further illustrated, with a motivating toy example, in Section 3.1. We will show, in the results section, that the proposed method produces better performances than an existing method [13] for higher order MRFs in localizing an object and than RANSAC for model estimation.

## 2 MRF formulation

Suppose there are $n$ datapoints $\{v_i|\ 1 \leq i \leq n\}$ that we wish to separate into two categories $A$ and $B$. We wish to label these samples using binary variables $\{x_i \in \{0,1\}|\ 1 \leq i \leq n\}$ where $x_i = 1$ implies $v_i \in A$ and $x_i = 0$ implies $v_i \in B$. Throughout the paper, we use the term 'group' to imply (interchangeably) a subset of $\{v_{i_1}, \ldots, v_{i_k}\}$ of size $k$ where $1 \leq i_l \leq n$ and $1 \leq l \leq k$. The likelihood that all members of $\{v_{i_1}, \ldots, v_{i_k}\}$ belong to $A$ is denoted by $\lambda_1(v_{i_1}, \ldots, v_{i_k})$ and

that they belong to $B$ by $\lambda_0(v_{i_1}, \ldots, v_{i_k})$. The term 'weight' is used interchangeably with likelihood measures in this paper.

As already stated, the objective of this work is to propose an algorithm to label the datapoints $v_i,\ 1 \leq i \leq n$ mainly based on the information provided in terms of small groups of them. We may or may not have some information about individual sample $v_i$ or pair of samples, but the focus is on how to utilize the knowledge we have for the subsets. To this end, the input to the algorithm is a set of small groups or subsets of data samples along with their likelihood values. Let $\mathcal{V}^k$ is the set of all groups $\{v_{i_1}, \ldots, v_{i_k}\}$ that satisfy a certain condition[2].

To establish a neighborhood $\mathcal{N}_i^k$ for any datapoint $v_i$, we define $v_j$ is a neighbor of $v_i$ if both $v_i$ and $v_j$ are members of any group $V^k \in \mathcal{V}^k$.

$$\mathcal{N}_i^k = \{\ v_j \mid j \neq i \text{ and } \ \exists_{V^k} \ \ \{v_i, v_j\} \subseteq V^k\}. \tag{1}$$

For inputs with different size groups, $k_1, k_2, \ldots$, we can similarly define a neighborhood system for each subset size.

Now, we can define a Markov Random Field (MRF) over the label variables $x_i$ assuming the Markov property that the value of $x_i$ depends on $x_j$ only if $v_j$ is a neighbor of $v_i$ [4]. Any subset $V^k \in \mathcal{V}^k$ defines a clique of size $k$ in the neighborhood system. Also, let $X^k$ denote the labels of members of $V^k$, i.e., $X^k = \{x_{i_1} \ldots x_{i_k}\}$ and $X$ denote labels of all $n$ datapoints $\{x_1, \ldots, x_n\}$.

It is well known that the probability of any assignment $p(X)$ depends on what is known as the Gibbs energy function $E(X)$ [3]. The energy function $E(X)$ is the summation of potential functions defined on cliques. Let $E^k(X^k)$ denote an appropriately defined clique potential function for the clique $V^k = \{v_{i_i}, \ldots, v_{i_k}\}$ of size $k$. With different size cliques, the Gibbs energy function equals to sum over all energy function of different sizes [3].

$$E(X) = \sum_{k=1}^{K} \sum_{V^k \in \mathcal{V}^k} E^k(X^k) \tag{2}$$

The optimal assignment $X = \{x_1, \ldots, x_n\}$ should minimize this Gibbs energy function. Most of the past studies determine the optimal configuration of $x_i$'s by minimizing $E(X)$ with maximum clique size of $K = 2$.

In this work, we formulate the energy function in Equation 2 as a submodular function [15]. Submodular functions are widely regarded as discrete analog of convex functions with diminishing returns. This kind of functions are theoretically guaranteed to be minimized in strongly polynomial time [5]. There are practical algorithms for solving MRFs with $K = 2$ in the literature [14]. There also exist linear programming solution for minimization of locally defined submodular functions, i.e., for solving MRF with $K > 2$; see Section 4.2 for references.

In next two sections, we try to motivate a novel energy function to solve the problem at hand and discuss what is necessary to render $E(X)$ as a submodular

[2] For example, a condition check retains only groups with weights larger (or errors less) than a threshold $\delta^k$.

function. In addition to suggesting a family of potential clique function for this purpose we show that for some special cases where there are very simple method to minimize the energy function.

## 3 Submodular Clique Potential

**Definition of Submodular Function:** Let $Y$ be a subset of $W$ and $\mathcal{P}(W)$ denotes set of all subsets of $W$. A function $f(\cdot)$ (defined over $\mathcal{P}(W)$) is called submodular if for any $u, z \in W \setminus Y$ the increase $f(Y \cup \{u\}) - f(Y)$ is larger than $f(Y \cup \{u, z\}) - f(Y \cup \{u\})$ [15]. This implies the rate of change of $f(Y)$ decreases as we keep augmenting the set $Y$ with new samples from $W \setminus Y$.

$$f(Y \cup \{u\}) - f(Y) \geq f(Y \cup \{u, z\}) - f(Y \cup \{u\}). \tag{3}$$

**Energy function as submodular function:** Now, our goal is to show that the energy function defined in Equation 2 is a submodular function. Observe that $E(X)$ in equation (2) is expressed as a summation of clique potentials $E^k$ defined over $\mathcal{P}(V^k)$. Form the study of Lovasz [16], we can define a function $\bar{E}^k(X)$ which enables $E^k(X^k)$ to handle the inputs from domain $\mathcal{P}(V)$, where $V = \{v_1, \ldots, v_n\}$. If we can show that $E^k(X^k)$ is submodular and replace them by $\bar{E}^k(X)$ in Equation 2, the Gibbs energy function becomes a summation of submodular functions. The nonnegative summation of submodular functions is also submodular [15]. Hence, if we could only compose a submodular clique potential $E^k(X^k)$, the energy function $E(X)$ becomes submodular.

### 3.1 Motivation for a New Clique Function

In our present scenario, we have weights associated with each data group that implies the likelihood that all members of the group belong to either of the two categories. We need a clique potential function whose value increases (or to be precise, not decrease) with the number of disagreement of the values of $x_{il}, 1 \leq l \leq k$, within a clique. This characteristic of a clique potential function would encourage all the datapoints to acquire same labels when we try to minimize the Gibbs energy function in Equation 2.

Now, we wish to handle the situations where the higher order weights give us decisive information (e.g., geometry, model conformation). Pointwise weights are either absent or are unable to offer a complete picture of the situation. Let us first examine whether or not clique functions of the existing works can solve the problems that we are discussing. Authors of [13] proposed a robust higher order submodular potential that was shown to produce excellent results for image segmentation using higher order cliques [12]. Let $\eta_0$ and $\eta_1$ denote the number of the datapoints in any subset $V^k$ take labels 0 and 1 respectively. The higher order clique function defined in [13] is the truncated minimum of a non-decreasing concave function $\mathcal{F}_c(\cdot)$ of $k - \eta_c$, where $c \in \{0, 1\}$ in our context,

$$E^k_{smooth}(X^k) = \min\{\min_{c \in \{0,1\}} \mathcal{F}_c(k - \eta_c), \gamma_{max}\}. \tag{4}$$

| Cost | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $v_5$ |
|---|---|---|---|---|---|
| $E^1(1)$ | 0.1 | 0.1 | 0.1 | 0.45 | 0.45 |
| $E^1(0)$ | 0.9 | 0.9 | 0.9 | 0.55 | 0.55 |

**Table 1.** Pointwise costs for toy txample.

| Weight | $\{v_1, v_2, v_3\}$ | $\{v_1, v_4, v_5\}$ | $\{v_2, v_4, v_5\}$ |
|---|---|---|---|
| $\lambda_1$ | 0.9 | 0.2 | 0.2 |
| $\lambda_0$ | 0.1 | 0.8 | 0.8 |

**Table 2.** Groupwise weights for toy txample.

However, the paper [13] later concentrates on a simpler form for $\mathcal{F}_c$ where it varies linearly between $[\gamma_c, \gamma_{max}]$, with $c \in \{0, 1\}$. In what follows, we provide an example scenario where this potential can not generate a non-trivial labeling.

**Toy Example:** Let us assume a toy dataset with 5 samples that we wish to label as 0 or 1. The energy to be minimized to produce this labeling comprise only pointwise and triplet-wise costs. The penalties or costs to assign any sample to a specific class is listed in Table 1. Three groups of size $k = 3$ are sampled from this dataset and their *weights* (as defined in first paragraph of Section 2) are given in Table 2. These weights will be utilized to compute the cost as defined in Equation 4 (which is different from the cost suggested by the proposed clique function defined later in Section 3.2).

The pointwise costs in Table 1 indicate that the first three samples $v_1, v_2$ and $v_3$ should be labeled as category $A$ since their $E^1(1)$ costs are much lower than $E^1(0)$. However, $E^1(\cdot)$ costs are not decisive for labels of $v_4$ and $v_5$. Table 2, on the other hand, tells us that $v_1, v_2$ and $v_3$ should be put in the category $A$ since the weight $\lambda_1$ is much larger than $\lambda_0$. But, the datapoints $v_4$ and $v_5$ should not be labeled the same as $v_1$ and $v_2$ since the $\lambda_1$ weights for the triples are much lower than corresponding $\lambda_0$.

Observe, first of all, that without the pointwise penalties, the clique function defined in Equation 4 will always produce trivial labeling, i.e., it will assign all datapoints to $c* = \arg\min_{c\in\{0,1\}} \mathcal{F}_c$. For linear form of $\mathcal{F}_c$, the label will be the one corresponding to minimum $\gamma_c$. The min operator over $\mathcal{F}_c$ plays the pivotal role for deciding the labels in this case. Therefore, the result would not change even if we use a non-linear $\mathcal{F}_c$.

The clique function $E^k_{smooth}$ of [13] generates a trivial solution even when the pointwise costs are incorporated. Let us use the linear form[3] of $\mathcal{F}_c = \gamma_c + \frac{\gamma_{max}-\gamma_c}{k}(k - \eta_c)$ with parameters $\gamma_c = 0$ and $\gamma_{max} = \lambda_1$. The resulting label will be all ones, 1, 1, 1, 1, 1. Due to the min operators, the result does not change if we use non-liner $\mathcal{F}_c$ or constant $\gamma_{\max} = 1$ or if we use $\gamma_{max} = \lambda_0$ for $\mathcal{F}_0$ and $\gamma_{max} = \lambda_1$ for $\mathcal{F}_1$. Changing parameter values would not be able to produce a non-trivial solution either with this clique function. This higher order clique function $E^k_{smooth}$ was designed as a smoothness function in [12, 13]. It is not suitable for situations where we only have higher order information about the data, e.g. model estimation from noisy data. Moreover, in presence of both pointwise and higher order costs/likelihoods, apparently it cannot fully utilize the higher order information [4]. However, it motivates us to propose the following clique function.

[3] This is how the authors of [13] defined $\gamma_{\max}$ in [12].

[4] We show one more interesting toy example in supplementary material where the clique function in Equation 4 produces trivial solution.

### 3.2 Proposed Clique function:

We define a clique potential function $E^k(\cdot)$ for clique $V^k$ as a linear combination of functions of $\eta_0$ and $\eta_1$.

$$E^k(X^k) = \beta_1 \lambda_1(V^k) g_1(k-\eta_1) + \beta_0 \lambda_0(V^k) g_0(k-\eta_0). \quad (5)$$

In this definition, $\lambda_1(V^k)$ and $\lambda_0(V^k)$ quantify the likelihood measures that all $v_{i_l} \in V^k$ belong to $A$ and $B$ respectively. $\beta_1$ and $\beta_0$ are two nonnegative parameters. The two functions $g_1(\cdot)$ and $g_0(\cdot)$ are non-decreasing functions. With high likelihood $\lambda_1(V^k)$ value for category $A$ with respect to that for $B$, the clique potential would be prone to increase $\eta_1$ and tolerate a small penalty $\lambda_0(V^k)\ g_0(k-\eta_0)$ as $\lambda_0(V^k)$ is small. In cases where the information $\lambda_0(V^k)$ is not available, we assume that the $\lambda_1(V^k)$ are normalized to $[0,1]$ and set $\lambda_0(V^k) = 1 - \lambda_1(V^k)$.

If we apply the proposed clique function $E^k$ with $g_c(k-\eta_c) = \frac{k-\eta_c}{k}$ on the toy example, the resulting label is $1,1,1,0,0$ respectively for the 5 datapoints (with or without pointwise penalties). This labeling perfectly confroms with the pointwise costs and triple likelihoods that we have in Tables 1 and 2. An additive combination of two functions $g_1$ and $g_0$ avoids trivial solutions for the problem we are dealing with. We will show examples of practical scenarios where the proposed clique is advantageous over the smoothness clique functions of [12, 13] in results section. As this clique function is new, we investigate, in the next subsection, what properties make it submodular. We show that, the clique potential is submodular whenever $g_c(\cdot)$ are either linear or concave. Furthermore, we also show that a combination of convex and concave functions for $g_1(\cdot)$ and $g_0(\cdot)$ is also submodular provided $g_1(\cdot)$ and $g_0(\cdot)$ satisfy some condition.

### 3.3 Properties Necessary for $E^k(X^k)$ to be Submodular

Recall that we want $g_c()$, $c \in \{0,1\}$, functions to be monotonically increasing. Let us express the proposed clique function of Equation 5 as a set function of the form given in Equation 3. Recall that $X^k$ is an indicator vector for set $A \in V^k$ (Section 2). We will write $E^k(X^K)$ as $f(A) = C_1 g_1(\eta_0) + C_0 g_0(\eta_1)$ where $C_1 = \beta_1 \lambda_1(V^k)$, $C_0 = \beta_0 \lambda_0(V^k)$ and $\eta_0 + \eta_1 = k$. If we add any $v_{i_l} \in V^k \setminus A$ to $A$, the clique function becomes $f(A \cup \{v_{i_l}\}) = C_1 g_1(\eta_0 - 1) + C_0 g_0(\eta_1 + 1)$. If we augment $A$ further by $v_{i_j} \in V^k \setminus (A \cup \{v_{i_l}\})$, the clique function becomes $f(A \cup \{v_{i_l}, v_{i_j}\}) = C_1 g_1(\eta_0 - 2) + C_0 g_0(\eta_1 + 2)$. To prove submodularity of $f(A)$, we have to prove that the successive increase in $f(A)$ diminishes.

$$f(A \cup \{v_{i_l}\}) - f(A) \geq f(A \cup \{v_{i_l}, v_{i_j}\}) - f(A \cup \{v_{i_l}\}) \quad (6)$$

$$\Rightarrow -C_0\ g_0\ (\eta_1 + 2) + 2\ C_0\ g_0(\eta_1 + 1) - C_0\ g_0(\eta_1) \geq C_1\ g_1(\eta_0) - 2\ C_1\ g_1(\eta_0 - 1) + C_1\ g_1(\eta_0 - 2) \quad (7)$$

$$\Rightarrow C_0 \quad \Big[g_0\ (\eta_1 + 2) - g_0(\eta_1 + 1)\Big] - \Big[g_0(\eta_1 + 1) - g_0(\eta_1)\Big] + C_1 \quad \Big[g_1(\eta_0) - g_1(\eta_0 - 1)\Big] - \Big[g_1(\eta_0 - 1) - g_1(\eta_0 - 2)\Big] \quad \leq 0 \quad (8)$$

Denoting the second order difference as $\Delta_c(n) = \big[g_c(n) - g_c(n-1)\big] - \big[g_c(n-1) - g_c(n-2)\big]$ where $c \in \{0, 1\}$, we observe that the condition for submodularity of the proposed clique function is as follows.

$$C_1 \ \Delta_1(\eta_0) \ + \ C_0 \ \Delta_0(\eta_1 + 2) \leq 0. \tag{9}$$

There are several options for $g_c$ functions to render the clique potential to be submodular.

**Linear $g_c(\cdot)$:** It is obvious that for linear $g_c$, the second order differences is 0 and the clique function is submodular[5].

**Concave $g_c(\cdot)$:** As the second derivative of any concave function is negative, the condition in Equation 9 holds.

**Constant $\Delta_c$:** For $g_c(\cdot)$ functions with constant $\Delta_c(n) = \bar{\Delta}_c$, one can design a submodular clique function with a concave $g_1(\cdot)$ and convex (or linear) $g_0(\cdot)$ whenever $C_1 > C_0$ and $|\bar{\Delta}_1| > |\bar{\Delta}_0|$ where $|\cdot|$ implies absolute value (and vice versa). We can also use a combination only when $C_1 > C_0$ and use concave functions for both $g_1()$ and $g_0()$ otherwise.

The linear form for $g_c(\cdot)$ was used for all the experiments in this paper. Further research is necessary to assess the applicability and effectiveness of the other two options. Finally, it is worth noting that the form of clique function with concave $g_c(\cdot)$ is substantially different from that proposed in Theorem 1 of [11]. The objective of [11] was to transform higher order interaction into pairwise ones so that one can apply Graph-Cut [14] method for inference. The authors of [11] prove (in Theorem 1 of the paper) that higher order clique functions defined as concave function over summation of pairwise functions (between samples within the clique) are submodular. We are showing that summation of concave functions defined over the whole clique is submodular. Due to Jensen's inequality, these two functions will not be equivalent even if we are able to find submodular pairwise functions for clique function in Equation 15 in [11].

We are not interested in using Graph-Cut for our inference because it is not clear yet how a general submodular higher order clique function can be reduced efficiently to pairwise interactions (see [17, 18]) [6]. Therefore, defining potentials in terms of pairwise functions is not necessary. Furthermore, it seems easier to design $g_c$ functions over whole clique than to design pairwise functions within the clique and then discover a concave function to be applied over their summation.

## 4 MRF Inference

The inference algorithm to be used to minimize the energy function $E(X)$ depends on the choice of $g_1(\cdot)$ and $g_0(\cdot)$ functions. The following two subsections describes two algorithms that can solve the inference problem. Again, these optimization methods are are well known in the literature.

[5] In fact, for linear $g_c$, the clique potential becomes modular

[6] There have been some recent works on how to reduce pseudo-boolean functions to pairwise interactions, see [19] and references therein, without connecting it to submodular functions

### 4.1 Special case: Linear $g_c(\cdot)$

The linear form of $g_c(\cdot),\ c \in \{0, 1\}$ increases from $l_c$ to $h_c$ in proportion to $k-\eta_c$.

$$g_c(k - \eta_c) = l_c + \frac{h_c - l_c}{k}(k - \eta_c). \tag{10}$$

Intuitively, an increase of $l_c$ or a decrease of $h_c$ will lower the number of datapoints to be labeled as category $c$ since the former would introduce a penalty to assign label $c$ to any sample in a subset and the latter would simply reduce the penalty to assign $1 - c$. It is straightforward to see that the value of $x_i$ is simply the label for which the summation of likelihood ratios weighted by the slope of $g_c(\cdot)$ is minimum (refer to supplementary material for details).

$$x_i^* = \arg \min_{c \in \{0,1\}} \sum_{V^k \in \mathcal{V}^k \wedge v_i \in V^k} \beta_c \lambda_c(V^k)\ \frac{h_c - l_c}{k}. \tag{11}$$

This solution can be computed in $O(n + |\mathcal{V}^k|)$, with one pass over all the subset weights and another pass over all the datapoints. For multiple $k$, the corresponding weights for each tuple size will be added. We used this simple inference algorithm in all our experiments. We do not need to apply other standard methods like max-flow min-cut algorithm for this simple decision though they are capable of solving it.

### 4.2 General Case: Nonlinear $g_c(\cdot)$

For nonlinear $g_c(\cdot),\ c \in \{0, 1\}$, the likelihood weights can not be evenly distributed to each of the datapoints. Problems with nonlinear $g_c(\cdot)$ can be solved using a linear programming formulation suggested by Cooper [20] for valued constraint satisfaction problem where the constraints are locally defined submodular functions. A linear programming relaxation for MRF inference for $K = 2$ was first suggested by Schlesinger in [21]. Since then, several algorithms have been proposed to solve this problem with $K = 2$ efficiently, see [22] for a review. However, Cooper's formulation [20] can also work with $K > 2$.

## 5 Experiments and Results

### 5.1 Model Estimation

Let us suppose that we have $n$ datapoints $v_i, i = 1, \ldots, n$, in some feature space. Part of these data samples are generated from some model whose parameters are unknown to us. We wish to estimate the model from these samples. In our approach, we sample $T^k$ subsets of size $k$ from the available datapoints. Then we try to fit the candidate model on each of these subsets using a least square method and compute the model estimation error (for each subset). We choose the value of $k$ larger than $s$ which is the minimum number of points required to fit a model (e.g. $s = 2$ to fit a line) such that the estimation problem is overconstrained. The subsets producing an error less than a problem specific threhold

| dataset | images | matches | Proposed missed | Proposed fp | RANSAC missed | RANSAC fp |
|---|---|---|---|---|---|---|
| Line | - | $67 + 80$ | $\mathbf{0.07 \pm 0.0305}$ | $\mathbf{0.07 \pm 0.0128}$ | $0.14 \pm 0.0372$ | $0.09 \pm 0.0042$ |
| Corridor | $\{bt.000, bt.010\}$ | $50 + 100$ | $\mathbf{0.03 \pm 0.0139}$ | $\mathbf{0.14 \pm 0.0181}$ | $0.07 \pm 0.0235$ | $0.17 \pm 0.0181$ |
| Valbonne | $\{000, 010\}$ | $30 + 60$ | $\mathbf{0.06 \pm 0.0157}$ | $\mathbf{0.13 \pm 0.0212}$ | $0.08 \pm 0.0254$ | $0.19 \pm 0.0205$ |
| Merton II | $\{002, 003\}$ | $50 + 100$ | $0.06 \pm 0.0160$ | $\mathbf{0.32 \pm 0.0105}$ | $\mathbf{0.01 \pm 0.0221}$ | $0.47 \pm 0.0138$ |
| Library | $\{002, 003\}$ | $50 + 100$ | $\mathbf{0.02 \pm 0.0121}$ | $\mathbf{0.17 \pm 0.0168}$ | $0.03 \pm 0.0248$ | $0.27 \pm 0.0208$ |

**Table 3.** Performance comparison for Model Estimation. First row: result for synthetic data. Other rows, first and second columns: datasets and images used. Third column: number of true matches + the number of incorrect matches. Fourth column: mean+standard deviation of missed inliers-Proposed. Fifth column: mean+standard deviation of false positives (fp)-Proposed. Sixth and Seventh: Those for RANSAC

$\delta^k$ constitutes the set $\mathcal{V}^k$. The error $\epsilon$ of any member $V^k \in \mathcal{V}^k$ is transformed to a likelihood measure by $\lambda_1(V^k) = \exp(-\epsilon/\sigma)$. For $\lambda_0(V^k)$ we use $1 - \lambda_1(V^k)$. Obviously, methods of [12, 13] can not be used for this model estimation problem since there are not pointwise costs. It is also not straightforward to design the pairwise interactions to apply clique function of [11].

**Synthetic data:** We plotted a line submerged in 55% noise samples. For the proposed method, we sampled $T^k = 2000$ subsets of size $k = 3$ and used $\delta^k = 0.5$, linear $g_c(\cdot)$, and $[h_1, l_1, h_0, l_0] = [1.01, 0.2, 1.0, 0]$. The model was also estimated using RANSAC [1] for comparison. We randomly chose pair of data samples for a maximum of 2000 times and used a threshold of $\delta_{ransac} = 0.4$ (value that produced minimum error) to determine inliers[7] from the whole dataset w.r.t the estimated model. The first row of Table 3 shows the mean and standard deviations of missed line points and false positive detections of both our algorithm and RANSAC over 100 runs.

**Fundamental matrix computation:** We have also tested our method for fundamental matrix computation. Given two images of the same scene from different viewpoint and camera location, the objective is to find the point matches that conform with the camera model expressed by the fundamental matrix. Four pairs of images from the Corridor, Valbonne, Merton II, Library datasets of the standard Oxford database[8] were used. We selected the two images with the largest variation in viewpoint, usually the first and last images (see Table 3).

A matlab implementation[9] to compute the fundamental matrix was used (keeping parameters values unchanged). Due to large variation in the image pairs, this code generated almost 0 correct matches. Therefore, we added a number of correct matches into the candidate set of matches to be considered for fundamental matrix estimation (also indicated in Table 3) and removed the overlap between the actual matches injected and the ones detected by Kovesi's code. For the proposed method each match is considered as a datapoint. We sampled 5000 subsets of size $k = 8$ (we know $s = 7$ in this case). Other parameter values are, threshold $\delta^k = 0.9$, and linear parameters $[h_1, l_1, h_0, l_0] = [1.01, 0.02, 1.0, 0]$ for clique function. For RANSAC, we sampled a subset of size 7 for at most 5000 times and used $\delta_{ransac} = 0.002$ as distance threshold (that produces the best result).

Table 3 shows the image names, number of true and detected candidate matches and the statistics for fraction of missed inliers and ratio of false posi-

[7] An inlier is a datapoint generated from (or, conforms with) the model.

[8] http://www.robots.ox.ac.uk/ vgg/data/data-mview.html

[9] Available online at http://www.csse.uwa.edu.au/ ~ pk/Research/MatlabFns/.

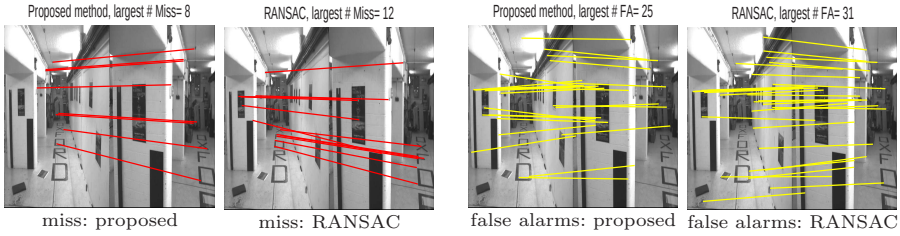


**Fig. 1.** Qualitative performances for detecting valid correspondences between two images. Red (yellow) line: matching point pair missed (falsely detected) by the method.

tive inliers in 100 runs for both the methods. As we can see in almost all the cases, proposed grouping method generates lower miss and false positive rates than that of RANSAC. The improved performances are primarily due to the fact that RANSAC starts with one subset of $s$ samples and keeps adding data samples greedily to refine the model. This greedy choice is not guaranteed to find the best model from the data. The proposed method tries to find which set of inliers agrees with the likelihood values of the most subsets of samples we are estimating from. In Figure 1, we show qualitative results of both proposed methods and RANSAC where they produced largest number of missed correct matches and false detections (miss+FA). This figure shows that in the worst case, the proposed method would miss less correct matches and generate fewer false alarms than those of RANSAC. In our experiments, we experienced same pattern for all the images.

### 5.2 Object localization

Part based representation of an object has gained much attention recent years in computer vision. Simple object part detectors will produce many detection outside the object region. We will apply the proposed grouping method to localize the object in an image by selecting correct object parts among these detected parts. THe candidate locations are spatially clustered Kadir-Brady salient region centers [23]. Each candidate location is a datapoint for the problem and the algorithm should detect which of these datapoints belong to the object. We use Geometric Blur descriptor [24] for these datapoints to create a codebook or bag of features (in a reduced dimensional feature space obtained through PCA).

We sample size $k$ subsets $\{v_{i_1}, \ldots, v_{i_k}\}$ of datapoints and represent them using parameters of geometric relation present among the members of the subset. Let $\pi$ be a permutation of $i_1, \ldots, i_k$ , which orders $v_{\pi(l)},\ 1 \leq l \leq k$ in increasing x-coordinate; $\zeta = [\zeta_l]_{l=1}^{k}$ where $\zeta_l$ is a code that represents $v_{\pi(l)}$; and $g$ be a vector of geometric parameters for the relation present in $v_{\pi(l)}$. Now, we wish to derive a probability for $g, \pi$ given object model description $\Theta_g^o, \Theta_a^o$ corresponding to geometric and appearance parameter for the object. The probability $p(g, \pi | \Theta_g^o, \Theta_a^o)$ can be expressed as a joint probability of $g$ and $\pi$ occurring together given the model.

$$p(g, \pi \mid \Theta_g^o, \Theta_a^o) = \sum_{\zeta} p(g, \pi \mid \zeta, \Theta_g^o, \Theta_a^o)\, p(\zeta \mid \Theta_a^o) = \sum_{\zeta} p(g \mid \zeta, \Theta_g^o)\, p(\pi \mid \zeta, \Theta_a^o)\, p(\zeta \mid \Theta_a^o).$$

We assume independence between geometric parameters $g$ and relative positions $\pi$ given ordered code representation $\zeta$, among $\pi_l,\ l = 1, \ldots, k$ given $\zeta_l$ and among

$\zeta_l$ themselves. Therefore, the above expression can be simplified as follows.

$$p(\pi|\ \zeta, \Theta_a^o) = \prod_{l=1}^{k} p(\pi_l|\ \zeta_l, \Theta_a^o) \ \text{ and } \ p(\zeta, |\ \Theta_a^o) = \prod_{l=1}^{k} p(\zeta_l|\ \Theta_a^o). \tag{12}$$

The probability $p(g \mid \zeta, \Theta_g^o)$ for geometric parameters are estimated by Kernel Density Estimation (KDE) with diagonal bandwidth matrix. The probability $p(\pi_l \mid \zeta_l, \Theta_a^o)$ is modeled using the feature descriptors $\phi(v_{\pi_l})$ and $\phi(\zeta_l)$ for $v_{\pi_l}$ and $\zeta_l$ respectively, i.e., $p(\pi_l \mid \zeta_l, \Theta_a^o)) \sim N(\phi(\zeta_l),\ 0.2I)$ where $I$ is the identity matrix. The priors$p(\zeta_l|\ \Theta_a^o)$ on the codes are computed empirically counting number of times $\zeta_l$ appeared in object.

An analogous model is learned for background subsets of parts. For each test image, subsets of size $k$ is collected the same way. The likelihood weight for object class $A$ is $\lambda_1(V^k) = p(g, \pi \mid \Theta_g^o, \Theta_a^o)$ and that for background class $B$ is $\lambda_0(V^k) = p(g, \pi \mid \Theta_g^b, \Theta_a^b)$ where $\Theta_g^b, \Theta_a^b$ describe the background model. A group will have a large likelihood weight $\lambda_1(V^k)$ if most of the codes representing the object parts have high prior probabilities and the geometric arrangement among them also has high likelihood value given object model. Given these weights, we run the proposed inference algorithm with linear $g_c(\cdot),\ c \in \{0, 1\}$ to decide which parts $v_i$ should have $x_i = 1$, i.e., which part should belong to the object.

The datasets used for this experiment are the Caltech cars (rear, with 126 images), motorbikes, airplanes images. Each dataset is split into two subsets to be used for training and testing. The training datasets are used to learn the codebook and object model probabilities. For both object and background parts, we identify the 3 nearest neighbors and sample subsets of sizes $k = 3$ respectively out of the part and its neighbors. The lengths of the sides of the triangle generated by subsets of size $k = 3$ (normalized by the perimeter) are used to encode the geometrical information among members of the group in addition to the internal angles.

We compared the results with a naive approach where the empirical probability of the code representing each part is used to determine its label. That is, the label $x_i$ of any $v_i$, is set to 1 if $p(\zeta_i|\Theta_a^o) > p(\zeta_i|\Theta_a^b)$, where $\zeta_i$ is the code representing $v_i$, and set to 0 otherwise. This method will be referred as Code Prior procedure hereafter.

Without an explicit model description, it is easy to realize that RANSAC can not be applied here. Furthermore, as we discussed before, the algorithms for higher order MRF inference in [13, 11, 12] can not be applied in this object model without pointwise costs. We utilized the the empirical code probability in order to employ the technique of [13] for Graph-Cut method with swap move in the present scenario with binary labels and only point-wise and triple-wise weights. Each location $v_i$ is considered as a node in the graph and its point-wise cost is inversely proportional to corresponding $p(\zeta_l|\Theta_a^o)$ and $p(\zeta_l|\Theta_a^b)$. The max weight for a 3-way clique, denoted by $\gamma_{\max}$ in [12, 13], is set to $\sum_\zeta p(g \mid \zeta, \Theta_g^o)\ p(\pi \mid \zeta, \Theta_a^o)$ for object class and $\sum_\zeta p(g \mid \zeta, \Theta_g^b)\ p(\pi \mid \zeta, \Theta_a^b)$ for background class. This conforms with how the authors of [13] defines $\gamma_{\max}$ to be proportional to a goodness measure of the corresponding clique in Equation 12 of [12]. The higher

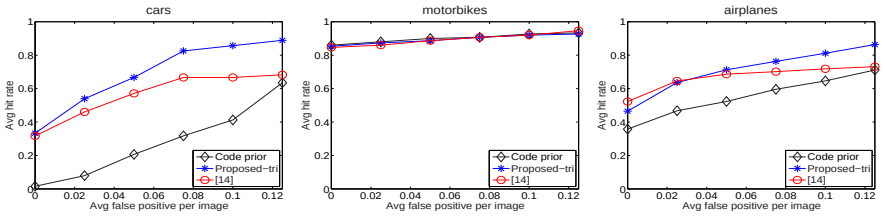


**Fig. 2.** Object localization rates for cars (left), motorbikes (middle) and airplanes (right) images.

order terms of [13] are distributed into members of the clique via an auxiliary variable. This is different from the manner that we break the higher order terms, compare Equation 11 of this article with Theorem 1 and Figure 3 of [13]. We used the parameters that produced best results for both the proposed method and for the method in [13].

For quantitative comparison, we plot the average number of images with a hit against the fraction of false detections allowed per image. If at least 5 of detected object parts fall within the object boundary of an image, we count it as one hit. The results are shown in Figure 2. It is quite natural that even using the code prior probability, and no geometric information, one can achieve good localization performance for motorbikes dataset as there are many images in this dataset where the object bounding box itself occupies more than 95% of the image area.

For the other two datasets, where the code prior probabilities are not sufficient to identify the object parts, the proposed method attains a higher hit rate with lower false positives than both the Code Prior approach and that of [13]. As discussed before, higher order information is not used for discrimination in [13]; it is rather used as smoothness constraint. On the other hand, the proposed method uses the higher order information of a group for discrimination. We are showing some qualitative results in Figure 3. Notice the zero detections of method [13] just as we anticipated before (see section 3.1).

## 6 Conclusion

This paper proposes a novel method for labeling datapoints primarily using higher order information generated given a model. We have formulated the problem as an MRF inference problem, proposed a clique function for this MRF so that the inference problem is tractable and identify a special case of the clique where there is a simple algorithm to solve the labeling problem. Our method has been shown to perform better than RANSAC for model estimation problem (where we know the analytic formm of the model). We also show a case where (the model is implicitly defined and) the proposed algorithm is advantageous over a previous higher order MRF algorithm for labeling task.

## References

1. Fischler, M.A., C., B.R.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. ACM **24** (1981) 381–395


**Fig. 3.** Qualitative object localization results- from top to bottom rows, the result for code prior, that of [13] and the proposed one respectively. The circles (filled with white) indicate the initially detected (spatially clustered) locations. The red * indicate the object parts detected by respective method.


2. Workshop: (25 years of ransac) website http://cmp.felk.cvut.cz/ransac-cvpr2006/, in conjunction with CVPR 2006.
3. Li, S.Z.: Markov Random Field Modeling in Image Analysis. Springer-Verlag (1991)
4. Geman, S., Geman, D.: Stochastic relaxation, gibbs distribution and bayesian restoration of images. PAMI **6** (1984) 721–741
5. Iwata, S., Fleischer, L., Fujishige, S.: A combinatorial strongly polynomial algorithm for minimizing submodular functions. Journal of ACM **48** (2001) 761–777
6. Aggarwal, S., Jongwoo, L., Zelnik-Manor, L., Perona, P., Kreigman, D.and Belongie, S.: Beyond pairwise clustering. In: CVPR. (2005)
7. Govindu, V.M.: A tensor decomposition for geometric grouping and segmentation. In: CVPR. (2005)
8. Zhou, D., J.H., Schlkopf, B.: Beyond pairwise classification and clustering using hypergraphs. Technical Report 143, Max Planck Institute for Biological Cybernetics (2005)
9. Shashua, A., Zass, R., Hazan, T.: Multi-way clustering using super-symmetric non-negative tensor factorization. In: ECCV. (2006)
10. Karypis, G., Kumar, V.: Multilevel k-way hypergraph partitioning. VLSI Design **11** (2000)
11. Kohli, P., Kumar, M., Torr, P.: $P^3$ and beyond: solving energies with higher order cliques. In: CVPR. (2007)
12. Kohli, P., Ladicky, L., Torr, P.: Robust higher order potentials for enforcing label consistency. In: CVPR. (2008)
13. Kohli, P., Ladicky, L., Torr, P.: Graph cuts for minimizing higher order potentials. In: Technical report, Oxford Brookes University (2008)
14. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. PAMI **23** (2001) 1222–1239
15. Fujishige, S.: Submodular Functions and Optimization. Elsevier Science Publishers (1991)
16. Lovasz, L.: Submodular functions and covnexity. Mathematical Programming-State of the Art (1983) 235–257
17. Freedman, D., Drineas, P.: Energy minimization via graph cuts: Settling what is possible. In: CVPR. (2005)
18. Živný, S., Jeavons, P.G.: Which submodular functions are expressible using binary submodular functions? Research Report RR-08-08, Oxford University Computing Lab, UK (2008)
19. Ishikawa, H.: Higher order clique reduction in binary graph cut. In: CVPR. (2009)
20. Cooper, M.C.: Minimization of locally defined submodular functions by optimal soft arc consistency. Constraints **13** (2008) 437–458
21. Schlesinger, M.I.: Syntactic analysis of two-dimensional visual signals in the presence of noise. Cybernetics and Systems Analysis **12** (1976) 612–628
22. Werner, T.: A linear programming approach to max-sum problem: A review. IEEE Trans. Pattern Anal. Mach. Intell. **29** (2007) 1165–1179

23. Kadir, T., Brady, M.: Saliency, scale and image description. IJCV **45** (2001) 83–105
24. Berg, A., Malik, J.: Geometric blur for template matching. In: CVPR. (2001)