Hypothesize and Bound: A Computational Focus of Attention Mechanism for Simultaneous N-D Segmentation, Pose Estimation and Classification Using Shape Priors

Diego Rother · Simon Schütz · René Vidal

Abstract Given the ever increasing bandwidth of the visual sensory information available to autonomous agents and other automatic systems, it is becoming essential to endow them with a sense of what is worthwhile their attention and what can be safely disregarded. This article presents a general mathematical framework to efficiently allocate the available computational resources to process the parts of the input that are relevant to solve a perceptual problem of interest. By solving a perceptual problem we mean to find the hypothesis \( H \) (i.e., the state of the world) that maximizes a function \( L(H) \), referred to as the evidence, representing how well each hypothesis “explains” the input. However, given the large bandwidth of the sensory input, fully evaluating the evidence for each hypothesis is computationally infeasible (e.g., because it would imply checking a large number of pixels). To address this problem we propose a mathematical framework with two key ingredients. The first one is a Bounding Mechanism (BM) to compute lower and upper bounds of the evidence of a hypothesis, for a given computational budget. These bounds are much cheaper to compute than the evidence itself, can be refined at any time by increasing the budget allocated to a hypothesis, and are frequently sufficient to discard a hypothesis. The second ingredient is a Focus of Attention Mechanism (FoAM) to select which hypothesis’ bounds should be refined next, with the goal of discarding non-optimal hypotheses with the least amount of computation.

The proposed framework has the following desirable characteristics: 1) it is very efficient since most hypotheses are discarded with minimal computation; 2) it is parallelizable; 3) it is guaranteed to find the globally optimal hypothesis or hypotheses; and 4) its running time depends on the problem at hand, not on the bandwidth of the input. In order to illustrate the general framework, in this article we instantiate it for the problem of simultaneously estimating the class, pose and a noiseless version of a 2D shape in a 2D image. To do this, we develop a novel theory of semidiscrete shapes that allows us to compute the bounds required by the BM. We believe that the theory presented in this article (i.e., the algorithmic paradigm and the theory of shapes) has multiple potential applications well beyond the application demonstrated in this article.

Keywords Focus of Attention · shapes · shape priors · hypothesize-and-verify · coarse-to-fine · probabilistic inference · graphical models · image understanding

1 Introduction

Humans are extremely good at extracting information from images. They can recognize objects from many different classes, even objects they have never seen before; they can estimate the relative size and position of objects in 3D, even from 2D images; and they can (in general) do this in poor lighting conditions, in cluttered scenes, or when the objects are partially occluded. However, natural scenes are in general very complex (Fig. 1), full of objects with intricate shapes, and so rich in textures, shades and details, that extracting all this information would be computationally very expensive, even for humans. Experimental psychology studies, on the other hand, suggest that (mental) computation is
In this article we propose a mathematical framework that uses a FoAM to allocate the available computational resources where they contribute the most to solve a given task. This mechanism is one of the parts of a novel family of inference algorithms that we refer to as hypothesize-and-bound (H&B) algorithms. These algorithms are based on the hypothesize-and-verify paradigm. In this paradigm a set of hypotheses and a function referred to as the evidence are defined. Each hypothesis \( H \) represents a different state of the world (e.g., which objects are located where) and its evidence \( L(H) \) quantifies how well this hypothesis “explains” the input image. In a typical hypothesize-and-verify algorithm the evidence of each hypothesis is evaluated and the hypothesis (or group of hypotheses) with the highest evidence is selected as the optimal.

However, since the number of hypotheses could be very large, it is essential to be able to evaluate the evidence of each hypothesis with the least amount of computation. For this purpose the second part of a H&B algorithm is a bounding mechanism (BM), which computes lower and upper bounds for the evidence of a hypothesis, instead of evaluating it exactly. These bounds are in general significantly less expensive to compute than the evidence itself, and they are obtained for a given computational budget (allocated by the FoAM), which in turn defines the tightness of the bounds (i.e., higher budgets result in tighter bounds). In some cases, these inexpensive bounds are already sufficient to discard a hypothesis (e.g., if the upper bound of \( L(H_1) \) is lower than the lower bound of \( L(H_2) \), \( H_1 \) can be safely discarded). Otherwise, these bounds can be efficiently and progressively refined by spending extra computational cycles on them (see Fig. 2). As mentioned above, the FoAM allocates the computational budget among the different hypotheses. This mechanism keeps track of the progress made (i.e., how much the bounds got closer to each other) for each computation cycle spent on each hypothesis, and decides-on-the-fly where to spend new computation cycles in order to economically discard as many hypotheses as possible. Because computation is allocated where it is most needed, H&B algorithms are in general very efficient; and because hypotheses are discarded only when they are proved suboptimal, H&B algorithms are guaranteed to find the optimal solution.

1.2 Application of H&B algorithms to vision

The general inference framework mentioned in the previous paragraphs is applicable to any problem in which bounds for the evidence can be inexpensively obtained for each hypothesis. Thus, to instantiate the framework to solve a particular problem, a specific BM has to be developed for that problem. (The FoAM, on the other hand, is common to many problems since it only communicates with the BM by allocating the computational budget to the different hypotheses and by “reading” the resulting bounds.)

In this article we illustrate the framework by instantiating it for the specific problem of jointly estimating the class and the pose of a 2D shape in a noisy 2D image, as well as recovering a noiseless version of this
shape. This problem is solved by “merging” information from the input image and from probabilistic models known a priori for the shapes of different classes.

As mentioned above, to instantiate the framework for this problem, we must define a mechanism to compute and refine bounds for the evidence of the hypotheses. To do this, we will introduce a novel theory of shapes and shape priors that will allow us to efficiently compute and refine these bounds. While still of practical interest, we believe that the problem chosen is simple enough to best illustrate the main characteristics of H&B algorithms and the proposed theory of shapes, without occluding its main ideas. Another instantiation of H&B algorithms, which we describe in [32], tackles the more complex problem of simultaneous object classification, pose estimation, and 3D reconstruction, from a single 2D image. In that work we also use the theory of shapes and shape priors to be described in this article to construct the BM for that problem.

1.3 Paper contributions

The framework we propose has several novel contributions that we group in two main areas, namely: 1) the inference framework using H&B algorithms; and 2) the shape representations, priors, and the theory developed around them. The following paragraphs summarize these contributions, while in the next section we put them in the context of prior work.

The first contribution of this paper is the use of H&B algorithms for inference in probabilistic graphical models. In particular, inference in graphical models containing loops. This inference method is not general, i.e., it is not applicable to any directed graph with loops. Rather it is specifically designed for the kinds of graphs containing pixels and voxels that are often used for vision tasks (which typically have a large number of variables and a huge number of loops among these variables). The proposed inference framework has several desirable characteristics. First, it is general, in the sense that the only requirement for its application is a BM to compute and refine bounds for the evidence of a hypothesis. Second, the framework is computationally very efficient, because it allocates computation dynamically where it is needed (i.e., refining the most promising hypotheses and examining the most informative image regions). Third, the total computation does not depend on the arbitrary resolution of the input image, but rather on the task at hand, or more precisely, on the similarity between the hypotheses that are to be distinguished. (In other words, “easy” tasks are solved very fast, while only “difficult” tasks require processing the image completely.) This allows us to avoid the common preprocessing step of downsampling the input image to the maximal resolution that the algorithm can handle, and permits us to use the original (possibly very high) resolution only in the parts of the image where it is needed. Fourth, the framework is fully parallelizable, which allows it to take advantage of GPUs or other parallel architectures. And fifth, it is guaranteed to find the globally optimal solution (i.e., the hypothesis with the maximum evidence), if it exists, or a set of hypotheses that can be formally proved to be undistinguishable at the maximum available image resolution. This guarantee is particularly attractive when the subjacent graphical model contains many loops, since existing probabilistic inference methods are either very inefficient or not guaranteed to find the optimal solution.

The second contribution relates to the novel shape representations proposed, and the priors presented to encode the shape knowledge of the different object classes. These shape representations and priors have three distinctive characteristics. First, they are able to represent a shape with multiple levels of detail. The level of detail, in turn, defines the amount of computation required to process a shape, which is critical in our framework. Second, it is straightforward and efficient to project a 3D shape expressed in these representations to the 2D image plane. This will be essential in the second part of this work [32] to efficiently compute how well a given 3D reconstruction “explains” the input image. And third, based on the theory developed for these shape representations and priors, it is possible to efficiently compute tight log-probability bounds. Moreover, the tightness of these bounds also depends on the level of detail selected, allowing us to dynamically trade computation for bound accuracy. In addition, the theory introduced is general and could be applied to solve many other problems as well.

1.4 Paper organization

The remainder of this paper is organized as follows. Section 2 places the current work in the context of prior relevant work, discussing important connections. Section 3 describes the proposed FoAM. To illustrate the application of the framework to the concrete problem of 2D shape classification, denoising, and pose estimation, in Section 4 we formally define this problem, and in Section 6 we develop the BM for it. In order to develop the BM, we first introduce in Section 5 a theory of shapes and shape priors necessary to compute the desired bounds for the evidence. Because the FoAM and the theory of shapes described in section 3 and 5 respectively, are general (i.e., not only limited to solve the problem described in Section 4), these sections were
written to be self contained. Section 7 presents experimental results obtained with the proposed framework, and Section 8 concludes with a discussion of the key contributions and directions for future research. In the continuation of this work [32], we extend the theory presented in this article to deal with a more complex problem involving not just 2D shapes, but also 3D shapes and their 2D projections.

2 Prior work

As mentioned in the previous section, this article presents contributions in two main areas: 1) the inference framework based on the FoAM; and 2) the shape representations proposed and the theory developed around them. For this reason, in this section we briefly review prior related work in these areas.

2.1 FoAM

Many computational approaches that rely on a focus of attention mechanism have been proposed over the years, in particular to interpret visual stimuli. These computational approaches can be roughly classified into two groups, depending on whether they are biologically inspired or not.

Biologically inspired approaches [9,39,36], by definition, exploit characteristics of a model proposed to describe a biological system. The goals of these approaches often include: 1) to validate a model proposed for a biological system; 2) to attain the outstanding performance of biological systems by exploiting some characteristics of these systems; and 3) to facilitate the interaction between humans and a robot by emulating mechanisms that humans use (e.g., joint attention [15]). Biological strategies, though optimized for eyes and brains during millions of years of evolution, are not necessarily optimal for current cameras and computer architectures. Moreover, since the biological attentional strategies are adopted at the foundations of these approaches by fiat (instead of emerging as the solution to a formally defined problem), it is often difficult to rigorously analyze the optimality of these strategies. In addition, these attentional strategies were in general empirically discovered for a particular sensory modality (predominantly vision) and are not directly applicable to other sensory modalities. Moreover, these strategies are not general enough to handle simultaneous stimuli coming from several sensory modalities (with some exceptions, e.g., [1]).

Since in this article we are mainly interested in improving the performance of a perceptual system (possibly spanning several sensory modalities), and since we want to be able to obtain optimality guarantees, we do not focus further on biologically inspired approaches.

The second class of focus of attention mechanisms contains those approaches that are not biologically inspired. Within this class we focus on those approaches that are not ad hoc (i.e., they are rigorously derived from first principles) and are general (i.e., they are able to handle different sensory modalities and tasks). This subclass contains at least two other approaches (apart from ours): Branch and Bound (B&B) and Entropy Pursuit (EP).

In a B&B algorithm [4], as in a H&B algorithm, an objective function is defined over the hypothesis space and the goal of the algorithm is to select the hypothesis that maximizes this function. A B&B algorithm proceeds by dividing the hypothesis space into subspaces, computing bounds of the objective function for each subspace (rather than for each hypothesis in the subspace), and discarding subspaces that can be proved to be non-optimal (because their upper bound is lower than that of some other subspace). In these algorithms computation is saved by evaluating whole groups of hypotheses, instead of evaluating each individual hypothesis. In contrast, in our approach the hypothesis space is discrete and bounds are computed for every hypothesis in the space. In this case computation is saved by discarding most of these hypotheses with very little computation. In other words, for most hypotheses the inexpensive bounds computed for the evidence of these hypotheses are enough to discard these hypotheses. As will be discussed in Section 8, this approach is complementary to B&B, and it would be beneficial to integrate both approaches into a single framework. Due to space limitations, however, this is not addressed in this article.

In an EP algorithm [11,38] a probability distribution is defined over the hypothesis space. Then, during each iteration of the algorithm, a test is performed on the input, and the probability distribution is updated by taking into account the result of the test. This test is selected as the one that is expected to reduce the entropy of the distribution the most. The algorithm terminates when the entropy of the distribution falls below a certain threshold. A major difference between EP and B&B/H&B algorithms is that in each iteration of EP a test is selected and the probability of each (of potentially too many) hypothesis is updated. In contrast, in each iteration of B&B/H&B, one hypothesis (or one group of hypotheses) is selected and only the bounds corresponding to it are updated. Unlike B&B and H&B algorithms, EP algorithms are not guaranteed to find the optimal solution.
A second useful criterion to classify computational approaches that rely on a FoAM considers whether attention is controlled only by “bottom-up” signals derived from salient stimuli, or whether it is also controlled by “top-down” signals derived from task demands, or from what a model predicts to be most relevant. Bottom-up approaches (e.g., [17]) are also known as data-driven approaches, while top-down approaches (e.g., [21]) are also known as task-driven approaches. Even though the significance of top-down signals in biological systems is well known, most current computer systems only consider bottom-up signals [9]. In contrast, all the three algorithmic paradigms described (H&B, B&B and EP), depending on the specific instantiation of these paradigms, are able to handle bottom-up as well as top-down signals. In particular, in the instantiation of H&B algorithms presented in Section 4, both kinds of signals are considered (in fact, it will be seen in Equation (59) that they play a symmetric role). In addition, in all the three algorithmic paradigms described above there is an explicit FoAM to control where the computation is allocated.

2.2 Inference framework

Many methods have been proposed to perform inference in graphical models [15]. Message passing algorithms are one class of these methods. Belief propagation (BP) is an algorithm in this class that is guaranteed to find the optimal solution in a loopless graph (i.e., a polytree) [2]. The loopy belief propagation (LBP) and the junction tree (JT) algorithms are two algorithms that extend the capabilities of the basic BP algorithm to handle graphs with loops. In LBP messages are exchanged exactly as in BP, but multiple iterations of the basic BP algorithm are required to converge to a solution. Moreover, the method is not guaranteed to converge to the optimal solution in every graph but only in some types of graphs [12]. In the JT algorithm [19], BP is run on a modified graph whose cycles have been eliminated. To construct this modified graph, the first step is “moralization,” which consists of marrying the parents of all the nodes. For the kinds of graphs we are interested in, however, this dramatically increases the clique size. While the JT algorithm is guaranteed to find the optimal solution, in our case this algorithm is not efficient because its complexity grows exponentially with the size of the largest clique in the modified graph.

Two standard “tricks” to perform exact inference (using BP) by eliminating the loops of a general graph are: 1) to merge nodes in the graph into a “supernode,” and 2) to make assumptions about the values of (i.e., to instantiate) certain variables, creating a different graph for each possible value of the instantiated variables (i.e., for each hypothesis) [25]. These approaches, however, bring their own difficulties. On the one hand, merging nodes results in a supernode whose number of states is the product of the number of states of the merged nodes. On the other hand, instantiating variables forces us to solve an inference problem for a potentially very large number of hypotheses.

In this work we propose a different approach to merge nodes that does not run into the problems mentioned above. Specifically, instead of assuming that the image domain is composed of a finite number of discrete pixels and merging them into supernodes, we assume that the image domain is continuous and consists of an infinite number of “pixels.” We then compute “summaries” of the values of the pixels in each region of the domain (this is formally described in Section 5). In order to solve the inference efficiently for each hypothesis, the total computation per hypothesis is trimmed down by using lower and upper bounds and a FoAM, as mentioned in Section 4.

2.3 Shape representations and priors

Since shape representations and priors are such essential parts of many vision systems, over the years many shape representations and priors have been proposed (see reviews in [8,6]). Among these, only a small fraction have the three properties required by our system and mentioned in Section 1.3, i.e., support multiple levels of detail, efficient projection, and efficient computation of bounds.

Scale space and orthonormal basis representations have the property that they can encode multiple levels of detail. In the scale-space representation [29], a shape (or image in general) is represented as a one-parameter family of smoothed shapes, parameterized by the size of the smoothing kernel used for suppressing fine-scale structures. Therefore, the representation contains a smoothed copy of the original shape at each level of detail. In the orthonormal basis representation, on the other hand, a shape is represented by its coefficients in an orthonormal basis. To compute these coefficients the shape is first expressed in the same representation as the orthonormal basis. For example, in [23] and [26] the contour of a shape is expressed in spherical wavelets and Fourier bases, respectively, and in [34] the signed distance function of a shape is written in terms of the principal components of the signed distance functions of shapes in the training database. The level of detail in this case is defined by the number of coefficients used to represent the shape in the basis. While these shape representations have the first
property mentioned above (i.e., multiple levels of detail), they do not have the other two, that is that it is not trivial to efficiently project 3D shapes expressed in these representations to the 2D image plane, or to compute the bounds that we want to compute.

The shape representations we propose, referred to as discrete and semidiscrete shape representations (defined in Section 3 and shown in Fig. 5) are respectively closer to region quadtrees/octrees [33] and to occupancy grids [7]. In fact, the discrete shape representation we propose is a special case of a region quadtree/octree in which the rule to split an element is a complex function of the input data, the prior knowledge, and the interaction with other hypotheses. Quadtrees and octrees have been previously used for 3D recognition [3] and 3D reconstruction [22] from multiple silhouettes (not from a single one, to the best of our knowledge, as we do in [32]). Occupancy grids, on the other hand, are significantly different from semidiscrete shapes since they store at each cell a qualitatively different quantity: occupancy grids store the posterior probability that an object is in the cell, while semidiscrete shapes store the measure of the object in the cell.

3 Focus of attention mechanism

In Section 1 we mentioned that a hypothesize-and-bound (H&B) algorithm has two parts: 1) a focus of attention mechanism (FoAM) to allocate the available computational cycles to the 2D image plane, to compute the bounds that we want to compute.

In Section 4 the evidence for our problem is shown to be related to the log-joint probability of the image $I$ and the hypothesis $H$. But again, from the point of view of the FoAM, it is not important how this function is defined; it only matters that hypotheses that “explain” the input better produce higher values. Thus, part of the goal of the FoAM is to select the hypothesis (or group of hypotheses) $H_i$ that best explain the input image, i.e.,

$$H_i = \arg \max_{H \in \mathcal{H}} L(H).$$

Now, suppose that the evidence $L(H_i)$ of a hypothesis $H_i$ is very costly to evaluate (e.g., because a large number of pixels must be processed to compute it), but lower and upper bounds for it, $L(H_i)$ and $\mathcal{T}(H_i)$, respectively, can be cheaply computed by the BM. Moreover, suppose that the BM can efficiently refine the bounds of a hypothesis $H_i$ if additional computational cycles (defined below) are allocated to the hypothesis. Let us denote by $L_n(H_i)$ and $\mathcal{T}_n(H_i)$, respectively, the lower and upper bounds obtained for $L(H_i)$ after $n_i$ computational cycles have been spent on $H_i$. If the BM is well defined, the bounds it produces must satisfy

$$L_{n_i+1}(H_i) \geq L_{n_i}(H_i), \quad \text{and} \quad \mathcal{T}_{n_i+1}(H_i) \leq \mathcal{T}_{n_i}(H_i),$$

for every hypothesis $H_i$, and every $n_i \geq 0$ (assume that $n_i = 0$ is the initialization cycle in which the bounds are first computed). In other words, the bounds must not become looser as more computational cycles are invested in their computation. Note that we expect different numbers of cycles to be spent on different hypotheses, ideally with “bad” hypotheses being discarded earlier than “better” ones (i.e., $n_1 < n_2$ if $L(H_1) \ll L(H_2)$).

The “computational cycles” mentioned above are our unit to measure the computational resources spent. Each computational cycle, or just cycle, is the computation that the BM spends to refine the bounds. While the exact conversion rate between cycles and operations depends on the particular BM used, what is important from the point of view of the FoAM is that all refinement cycles take approximately the same number of operations (defined to be equal to one computational cycle).

The full goal of the FoAM can now be stated as to select the hypothesis $H_i$ that satisfies

$$L_{n_j}(H_i) > \mathcal{T}_{n_j}(H_j) \quad \forall j \neq i,$$

while minimizing the total number of cycles spent, $\sum_{j=1}^{N_H} n_j$. If these inequalities are satisfied, it can be proved that $H_i$ is the optimal hypothesis, without having to compute exactly the evidence for every hypothesis (which is assumed to be much more expensive than just computing the bounds). However, it is possible that after all the hypotheses in a set $\mathcal{H}_i \subset \mathcal{H}$ have been refined to the fullest extent possible, their upper bounds are
still bigger than or equal to the maximum lower bound
\( \gamma \triangleq \max_{H \in \mathcal{H}} L_{\text{BM}}(H) \), i.e.,
\[
\mathcal{T}_{n_i}(H_i) \geq \gamma \quad \forall H_i \in \mathcal{H}.
\] (4)

In this situation all the hypotheses in \( \mathcal{H} \), could possibly be optimal, but we cannot say which one actually is. We just do not have the right input to distinguish between them (e.g., because the resolution of the input image is insufficient). We say that these hypotheses are indistinguishable given the current input. In short, the FoAM will terminate either because it has found the optimal hypothesis (satisfying (3)), or because it has found a set of hypotheses that are indistinguishable from the optimal hypothesis given the current input (and satisfies (4)).

These termination conditions can be achieved by very different budgets that allocate different number of cycles to each hypothesis. We are interested in finding the budget that achieves them in the minimum number of cycles. Finding this minimum is in general not possible since the FoAM does not "know," a priori, how the bounds will change for each cycle it allocates to a hypothesis. For this reason, we propose a heuristic to select the next hypothesis to refine at each point in time. Once a hypothesis is selected, one cycle is allocated to this hypothesis, which is thus refined once by the BM. This selection-refinement cycle is continued until termination conditions are reached.

According to the heuristic proposed, the next hypothesis to refine, \( H_{i_\ast} \), is chosen as the one that is expected to produce the greatest reduction \( \Delta P(H_{i_\ast}) \) in the following potential \( P \),
\[
P \triangleq \sum_{H_i \in \mathcal{A}} (\mathcal{T}_{n_i}(H_i) - \gamma),
\] (5)
where \( \mathcal{A} \) is the set of all the hypotheses not yet discarded (i.e., those that are active), \( \gamma \) is the maximum lower bound defined before, and \( n_i \) is the number of refinement cycles spent on hypothesis \( H_i \). This particular expression for the potential was chosen for two reasons: 1) because it reflects the workload left to be done by the FoAM; and 2) because it is minimal when termination conditions have been achieved.

In order to estimate the potential reduction \( \Delta P(H) \) expected when hypothesis \( H \) is refined (as required by the heuristic), we need to first define a few quantities (Fig. 2). We define the margin \( M_n(H) \) of a hypothesis \( H \) after \( n \) cycles have been spent on it, as the difference between its bounds, i.e., \( M_n(H) \triangleq \mathcal{T}_n(H) - L_n(H) \). Then we define the reduction of the margin of this hypothesis during its \( n \)-th refinement as \( \Delta M_n(H) \triangleq M_{n-1}(H) - M_n(H) \). It can be seen that this quantity is positive, and because in general early refinements produce larger margin reductions than later refinements, it has a general tendency to decrease. Using this quantity we predict the reduction of the margin in the next refinement using an exponentially weighted moving average, \( \Delta M_{n+1}(H) \triangleq \alpha \Delta M_n(H) + (1 - \alpha)M_n(H) \), where \( 0 < \alpha < 1 \). This moving average is initialized as \( \Delta M_0(H) = \beta M_0(H) \). (In this work we used \( \alpha = 0.9 \) and \( \beta = 0.25 \).)

The predicted potential reduction \( \Delta P(H) \) depends on whether the refinement of \( H \) is expected to increase \( \gamma \) or not: when \( \gamma \) increases, every term in (5) is reduced; when it does not, only the term corresponding to \( H \) is reduced (compare figures 2b and 2c). Let us assume that the reduction in the upper bound, \( \mathcal{T}_{n+1}(H) - \mathcal{T}_n(H) \), is equal to the increase in the lower bound, \( L_{n+1}(H) - L_n(H) \), which is therefore predicted to be equal to \( \Delta M_{n+1}(H)/2 \). Let us define \( \Delta \gamma(H) \) to be the increase in \( \gamma \) when \( H \) is refined, thus \( \Delta \gamma(H) \triangleq \max\{\mathcal{T}_n(H) + \Delta M_{n+1}(H)/2 - \gamma, 0\} \). Then the following expression is an estimate for the potential reduction when \( H \) is refined:
\[
\Delta P(H) = \begin{cases} 
\Delta M_{n+1}(H)/2 + |\Delta \gamma(H)|, & \text{if } \Delta \gamma(H) > 0 \\
\Delta M_{n+1}(H)/2, & \text{otherwise.}
\end{cases}
\] (6)
As mentioned before, the hypothesis that maximizes this quantity is the one selected to be refined next.

The algorithm used by the FoAM is thus the following (a detailed explanation is provided immediately afterwards):
1: \( \gamma \leftarrow -\infty \)
2: for \( i = 1 \) to \( N_H \) do
3: \[ [L(H_i), \mathcal{T}(H_i)] \leftarrow \text{ComputeBounds}(H_i) \]
4: if \( L(H_i) > \gamma \) then
5: \( \gamma \leftarrow L(H_i) \)
6: end if
7: if \( T(H_i) > \gamma \) then
8: \( \Delta M_0(H_i) \leftarrow \beta M_0(H_i) \)
9: Compute \( \Delta P(H_i) \)
10: A.Insert(\( H_i, \Delta P(H_i) \))
11: end if
12: end while

while not reached Termination Conditions do
13: \( H_i \leftarrow A.GetMax() \)
14: if \( T(H_i) > \gamma \) then
15: \( [L(H_i), T(H_i)] \leftarrow \text{RefineBounds}(H_i) \)
16: Compute \( \Delta P(H_i) \)
17: A.Insert(\( H_i, \Delta P(H_i) \))
18: if \( L(H_i) > \gamma \) then
19: \( \gamma \leftarrow L(H_i) \)
20: end if
21: end if
22: end if
23: end while

The first stage of the algorithm is to initialize the bounds for all the hypotheses (line 3 above), use these bounds to estimate the expected potential reduction for each hypothesis (lines 8-9), and to insert the hypotheses in the priority queue \( A \) using the potential reduction as the key (line 10). This priority queue \( A \), supporting the usual operations \text{Insert} and \text{GetMax}, contains the hypotheses that are active at any given time. The \text{GetMax} operation, in particular, is used to efficiently find the hypothesis that, if refined, is expected to produce the greatest potential reduction. During this first stage the maximum lower bound \( \gamma \) is also initialized (lines 1 and 4-6).

In the second stage of the algorithm, hypotheses are selected and refined alternately until termination conditions are reached. The next hypothesis to refine is simply obtained by extracting from the priority queue \( A \) the hypothesis that is expected to produce, if refined, the greatest potential reduction (line 14). If this hypothesis is still viable (line 15), its bounds are refined (line 16), its expected potential reduction is recomputed (line 17), and it is reinserted into the queue (line 18). If necessary, the maximum lower bound is also updated (lines 19-21). One issue to note in this procedure is that the potential reductions used as key when hypotheses are inserted in the queue \( A \) are outdated once \( \gamma \) is modified. Nevertheless this approximation works well in practice and allows a complete hypothesis selection/refinement cycle (lines 13-23) to run in \( O(\log |A|) \) where \(|A|\) is the number of active hypotheses. This complexity is determined by the operations on the priority queue.

Rother and Sapiro [11] have suggested a different heuristic to select the next hypothesis to refine. Their heuristic consists on selecting the hypothesis whose current upper bound is greatest. This heuristic, however, in general required more cycles than the heuristic we are proposing here. To see why, consider the case in which, after some refinement cycles, two active hypotheses \( H_1 \) and \( H_2 \) still remain. Suppose that \( H_1 \) is better than \( H_2 \) (\( T_{n_1}(H_1) \gg T_{n_2}(H_2) \)), but at the same time it has been more refined (\( n_1 \gg n_2 \)). As mentioned before, because of the decreasing nature of \( \Delta M \), in these conditions we expect \( \Delta M_{n_2}(H_2) \gg \Delta M_{n_1}(H_1) \). Therefore, if we chose to refine \( H_1 \) (as in [11]) many more cycles will be necessary to distinguish between the hypotheses than if we had chosen to refine \( H_2 \) (as in the heuristic explained above). However, the strategy of choosing the less promising hypothesis is only worthwhile when there are few hypotheses remaining, since computation in that case is invested in a hypothesis that is ultimately discarded. The desired behavior is simply and automatically obtained by minimizing the potential defined in \( \hat{\Delta P}(\cdot) \), and this ensures that computation is spent sensibly.

4 Definition of the Problem

The FoAM described in the previous section is a general algorithm that can be used to solve many different problems, as long as: 1) the problems can be formulated as selecting the hypothesis that maximizes some evidence function within a set of hypotheses, and 2) a suitable BM can be defined to bound this evidence function. To illustrate the use of the FoAM to solve a concrete problem, in this section we define the problem, and in Section 5 we derive a BM for this particular problem.

Given an input image \( I : \Omega \rightarrow \mathbb{R}^c \) \((c \in \mathbb{N}, c > 0)\) in which there is a single “shape” corrupted by noise, the problem is to estimate the class \( K \) of the shape, its pose \( T \), and recover a noiseless version of the shape. This problem arises, for example, in the context of optical character recognition [10] and shape matching [40]. For clarity it is assumed in this section that \( \Omega \subset \mathbb{Z}^2 \) (i.e., the image is composed of discrete pixels arranged in a 2D grid).

To solve this problem using a H&B algorithm, we define one hypothesis \( H \) for every possible pair \((K, T)\). By selecting a hypothesis, the algorithm is thus estimating the class \( K \) and pose \( T \) of the shape in the image. As we will later show, in the process a noiseless version of the shape will also be obtained.

In order to define the problem more formally, suppose that the image domain \( \Omega \) contains \( n \) pixels, \( x_1, \ldots, x_n \), and that there are \( N_K \) distinct possible shape classes,
each one characterized by a known shape prior $B_K$ ($1 \leq K \leq N_K$) defined on the whole discrete plane $\mathbb{Z}^2$, also containing discrete pixels. Each shape prior $B_K$ specifies, for each pixel $x' \in \mathbb{Z}^2$, the probability that the pixel belongs to the shape $q$, $p_{B_K}(x') \triangleq P(q(x') = 1|K)$, or to the complement of the shape, $P(q(x') = 0|K) = 1 - p_{B_K}(x')$. We assume that $p_{B_K}$ is zero everywhere, except (possibly) in a region $\Omega_K \subset \mathbb{Z}^2$ called the support of $p_{B_K}$. We will say that a pixel $x'$ belongs to the Foreground if $q(x') = 1$, and to the Background if $q(x') = 0$ (Foreground and Background are the labels of the two possible states of a shape for each pixel).

Let $T \in \{T_1, \ldots, T_{N_T}\}$ be an affine transformation in $\mathbb{R}^2$, and call $B_T$ (recall that $H = (K, T)$) the shape prior that results from transforming $B_K$ by $T$, i.e., $p_{B_T}(x) \triangleq P(q(x) = 1|H) \triangleq p_{B_K}(T^{-1}x)$ (disregard for the moment the complications produced by the misalignment of pixels). The state $q(x)$ in a pixel $x$ is thus assumed to depend only on the class $K$ and the transformation $T$ (in other words, it is assumed to be conditionally independent of the states in the other pixels, given the hypothesis $H$).

Now, suppose that the shape $q$ is not observed directly, but rather that it defines the distribution of a feature (e.g., colors, edges, or in general any feature) to be observed at a pixel. In other words, if a pixel $x$ belongs to the background (i.e., if $q(x) = 0$), its feature $f(x)$ is distributed according to the probability density function $p_x(f(x)|q(x) = 0)$, while if it belongs to the foreground (i.e., if $q(x) = 1$), $f(x)$ is distributed according to $p_x(f(x)|q(x) = 1)$ (the subscript $x$ in $p_x$ was added to emphasize the fact that the probability of observing a feature $f(x)$ at a pixel $x$ depends on the state of the pixel $q(x)$ and on the particular pixel $x$, or in other terms, $p_x(f_0|x_0) \neq p_x(f_0|x_0)$ if $x \neq y$ and $f_0$ and $q_0$ are two arbitrary values of $f$ and $q$, respectively). This feature $f(x)$ is assumed to be independent of the feature $f(y)$ and the state $q(y)$ in every other pixel $y$, given $q(x)$.

The conditional independence assumptions described above can be summarized in the factor graph of Fig. 3 (see [2] for more details on factor graphs). It then follows that the joint probability of all pixel features $f$, all states $q$, and the hypothesis $H = (K, T)$, is

$$p(f,q,H) = P(H) \prod_{x \in H} P_q(f(x)|q(x))P(q(x)|H). \quad (7)$$

Then, our goal can be simply stated as solving

$$\max_{q,H} p(f,q,H) = \max_{H \in \mathbb{H}} L''(H), \quad \text{with} \quad L''(H) \triangleq \max_q p(f,q,H). \quad (8)$$

Fig. 3 Factor graph proposed to solve our problem of interest. A factor graph, [2], has a variable node (circle) for each variable, and a factor node (square) for each factor in the system’s joint probability. Factor nodes are connected to the variable nodes of the variables in the factor. Observed variables are shaded. A plate indicates that there is an instance of the nodes in the plate for each element in a set (indicated on the lower right). The plate in this graph hides the existing loops. See text for details.

We could solve this problem naively by computing (9) for every $H \in \mathbb{H}$. However, to compute (9) all the pixels in the image (or at least all the pixels in the support of $\Omega_K$) need to be processed in order to evaluate the product in (7) (since the solution $q^*$ that maximizes (9) can be written explicitly). Because this might be very expensive, we need a BM to evaluate the evidence without having to process every pixel in the image.

Therefore, instead of using this naive approach, we will use an H&B algorithm to find the hypothesis $H$ that maximizes an expression simpler than $L''(H)$, that is equivalent to it (in the sense that it has the same maxima). This simpler expression is what we have called the evidence, $L(H)$, and will be derived from (9) in Section 6.1. Before deriving the evidence, however, in Section 5 we present the mathematical framework that will allow us to do that, and later to develop the BM for this evidence.

Fig. 3 Factor graph proposed to solve our problem of interest. A factor graph, [2], has a variable node (circle) for each variable, and a factor node (square) for each factor in the system’s joint probability. Factor nodes are connected to the variable nodes of the variables in the factor. Observed variables are shaded. A plate indicates that there is an instance of the nodes in the plate for each element in a set (indicated on the lower right). The plate in this graph hides the existing loops. See text for details.

5 A new theory of shapes

As mentioned before, H&B algorithms have two parts: a FoAM and a BM. The FoAM was already introduced in Section 3. While the same FoAM can be used to solve many different problems, each BM is specific to a particular problem. Towards defining the BM for the problem described in the previous section (that will be done in Section 6), in this section we introduce a mathematical framework that will allow us to compute the bounds.

To derive formulas to bound the evidence $L(H)$ of a hypothesis $H$ (the goal of the BM) for our specific prob-
lem of interest, we introduce in this section a framework to represent shapes and to compute bounds for their log-probability (in Section 6.1 we will show that this log-probability is closely related to the evidence). Three different shape representations will be introduced (Fig. 3). Continuous shapes are the shapes that we would observe if our cameras (or 3D scanners) had “infinite resolution.” In that case it would be possible to compute the evidence \( L(H) \) of a hypothesis with “infinite precision” and therefore always select the (single) hypothesis whose evidence is maximum (except in concocted examples which have very low probability of occurring in practice). However, since real cameras and scanners have finite resolution, we introduce two other shape representations that are especially suited for this case: \textit{discrete} and \textit{semidiscrete} shape representations.

Discrete shapes will allow us to compute a lower bound \( L(H) \) for the evidence of a hypothesis \( H \). Semidiscrete shapes, on the other hand, will allow us to compute an upper bound \( U(H) \) for the evidence of a hypothesis \( H \). Discrete and semidiscrete shapes are defined on partitions of the input image (i.e., non-overlapping regions that completely cover the image, see Fig. 4). Finer partitions result in tighter bounds, more computation, and the possibility of distinguishing more similar hypotheses. Coarser partitions on the other hand, result in looser bounds, less computation and more hypotheses that are indistinguishable (in the partition).

Previously we have assumed that the image domain, \( \Omega \subset \mathbb{R}^2 \), consisted on discrete pixels arranged in a 2D grid. For reasons that will soon become clear, however, we assume now from on that the image domain \( \Omega \subset \mathbb{R}^2 \) is continuous. Thus, to discretize this continuous domain we rely on “partitions,” defined next.

**Definition 1 (Partitions)** Given a set \( \Omega \subset \mathbb{R}^d \), a \textit{partition} \( \Pi(\Omega) = \{\Omega_1, \ldots, \Omega_n\} \) with \( \Omega_i \neq \emptyset \), is a disjoint cover of the set \( \Omega \) (Fig. 4). Formally, \( \Pi(\Omega) \) satisfies

\[
\bigcup_{i=1}^{n} \Omega_i = \Omega, \quad \text{and} \quad \Omega_i \cap \Omega_j = \emptyset \ \forall i \neq j.
\]  

(10)

A partition \( \Pi(\Omega) = \{\Omega_1, \ldots, \Omega_n\} \) is said to be \textit{uniform} if all the elements in the partition have the same measure \( |\Omega_i| = \frac{|\Omega|}{n} \) for \( i = 1, \ldots, n \). (Throughout this article we use the notation \( |\Omega| \) to refer to, depending on the context, the measure or the cardinality of a set \( \Omega \).) This measure is referred to as the \textit{unit size} of the partition.

Given two partitions \( \Pi^1(\Omega) \) and \( \Pi^2(\Omega) \) of a set \( \Omega \subset \mathbb{R}^d \), \( \Pi^2 \) is said to be finer than \( \Pi^1 \), and \( \Pi^1 \) is said to be coarser than \( \Pi^2 \), if every element of \( \Pi^2 \) is a subset of some element of \( \Pi^1 \) (Fig. 4). We denote this relationship as \( \Pi^2 \leq \Pi^1 \). Note that two partitions are not always comparable, thus the binary relationship “\( \leq \)” defines a partial order in the space of all partitions.

5.1 Discrete shapes

**Definition 2 (Discrete shapes)** Given a partition \( \Pi(\Omega) \) of the set \( \Omega \subset \mathbb{R}^d \), the \textit{discrete shape} \( \hat{S} \) (Fig. 5b) is defined as the function \( \hat{S} : \Pi(\Omega) \to \{0,1\} \).

**Definition 3 (Log-probability of a discrete shape)** Let \( \hat{S} \) be a discrete shape in some partition \( \Pi(\Omega) = \{\Omega_1, \ldots, \Omega_n\} \), and let \( \hat{B} = \{\hat{B}_1, \ldots, \hat{B}_n\} \) be a family of independent Bernoulli random variables referred to as a discrete Bernoulli field (BF). Let \( \hat{B} \) be characterized by the success rates \( p_B(i) \equiv P(\hat{B}_i = 1) \in (\varepsilon, \varepsilon) \) for \( i = 1, \ldots, n \), and \( 0 < \varepsilon \ll 1 \). To avoid the problems derived from assuming complete certainty, i.e. success rates of 0 or 1, following Cromwell’s rule \[21\], we will only consider success rates in the open interval \( (\varepsilon, 1-\varepsilon) \).

The \textit{log-probability} of a \textit{discrete shape} is defined as

\[
\log P(\hat{B} = \hat{S}) \equiv \sum_{i=1}^{n} \log P(\hat{B}_i = \hat{S}(\Omega_i)) = \sum_{i=1}^{n} \left[ (1 - \hat{S}(\Omega_i)) \log P(\hat{B}_i = 0) + \hat{S}(\Omega_i) \log P(\hat{B}_i = 1) \right]
\]

\[= Z_B + \sum_{i=1}^{n} \hat{S}(\Omega_i) \delta_B(i), \tag{11}\]
where $Z_B \triangleq \sum_{i=1}^{n} \log \left(1 - p_B(i)\right)$ is a constant and $\delta_B(i) \triangleq \log \left(p_B(i)/(1 - p_B(i))\right)$ is the logit function of $p_B(i)$.

The discrete BFs used in this work arise from two sources: background subtraction and shape priors. To compute a discrete BF $B_f$ using the Background Subtraction technique [22], recall the probability densities $p_{\Omega_i}(f(\Omega_i)|q(\Omega_i) = 0)$ and $p_{\Omega_i}(f(\Omega_i)|q(\Omega_i) = 1)$ defined in Section 4 to model the probability of observing a feature $f(\Omega_i)$ at a given pixel $\Omega_i$, depending on the pixel’s state $q(\Omega_i)$. The success rates of the discrete BF $B_f$ are thus defined as

$$p_{B_f}(i) \triangleq \frac{p_{\Omega_i}(f(\Omega_i)|q(\Omega_i) = 1)}{p_{\Omega_i}(f(\Omega_i)|q(\Omega_i) = 0) + p_{\Omega_i}(f(\Omega_i)|q(\Omega_i) = 1)}.$$  \hspace{1cm} (12)

To compute a discrete BF $B_s$ associated with a discrete shape prior, we can estimate the success rates $p_{B_s}(i)$ of $B_s$ from a collection of $N$ discrete shapes, $S = \{S_1, \ldots, S_N\}$, assumed to be aligned in the set $\Omega$. These discrete shapes can be acquired by different means, e.g., using a 2D or 3D scanner, for $d = 2$ or $d = 3$, respectively. The success rate $p_{B_s}(i)$ of a particular Bernoulli variable $B_s(i)$ is thus estimated from $\{S_1(\Omega_i), \ldots, S_N(\Omega_i)\}$ using the standard formula for the estimation of Bernoulli distributions [16].

$$p_{B_s}(i) = \frac{1}{N} \sum_{j=1}^{N} S_j(\Omega_i).$$  \hspace{1cm} (13)

Discrete shapes, as in Definition 2, have two limitations that must be addressed to enable subsequent developments. First, the log-probability in (11) depends (implicitly) on the unit size of the partition (which is related to the image resolution), preventing the comparison of log-probabilities of images acquired at different resolutions (this will be further explained in Definition 4). Second, it was assumed in (11) that the Bernoulli variables $B_s(i)$ and the pixels $\Omega_i$ were perfectly aligned. However, this assumption might be violated if a transformation (e.g., a rotation) is applied to the shape. To overcome these limitations, and also to facilitate the proofs that will follow, we introduce next the second shape representation, that of a continuous shape.

5.2 Continuous shapes

Definition 4 (Continuous shapes) Given a set $\Omega \subset \mathbb{R}^d$, we define a continuous shape $S$ to be a function $S: \Omega \to \{0, 1\}$ (Fig. 5b). We will often abuse notation and refer to the set $S = \{x \in \Omega : S(x) = 1\}$ as also the shape. To avoid pathological cases, we will require the set $S$ to satisfy two regularity conditions: 1) to be open (in the usual topology in $\mathbb{R}^d$) and 2) to have a boundary (as defined in [13]) of measure zero.

Given a discrete shape $\hat{S}$ defined on a partition $\Pi(\Omega) = \{\Omega_1, \ldots, \Omega_n\}$, the continuous shape $S(\Omega) \triangleq \hat{S}(\Omega_i) \forall x \in \Omega_i$, is referred to as the continuous shape produced by the discrete shape $\hat{S}$, and is denoted as $S \sim \hat{S}$ or $\hat{S} \sim S$. Intuitively, $S$ extends $\hat{S}$ from every element of $\Pi(\Omega)$ to every point $x \in \Omega$.

We would like now to extend the definition of the log-probability of a discrete shape (in Definition 3) to include continuous shapes. Toward this end we first introduce continuous BFs, which play in the continuous case the role that discrete BFs play in the discrete case.

Definition 5 (Continuous Bernoulli Fields) Given a set $\Omega \subset \mathbb{R}^d$, a continuous Bernoulli field (or simply a BF) is the construction that associates a Bernoulli random variable $B_x$ to every point $x \in \Omega$. The success rate for each variable in the field is given by the function $p_B(x) \triangleq P(B_x = 1)$. The corresponding logit function $\delta_B(x) \triangleq \log \left(\frac{p_B(x)}{1 - p_B(x)}\right)$ and constant term $Z_B \triangleq \int_{\Omega} \log (1 - p_B(x))dx$ are as in Definition 3.

We will only consider in this work functions $p_B(x)$ such that $|Z_B| < \infty$ and $\delta_B(x)$ is a measurable function [13]. Furthermore, since $\varepsilon < p_B(x) < 1 - \varepsilon \forall x \in \Omega$, $\delta_B(x) \in (-\delta_{\max}, \delta_{\max}) \forall x \in \Omega$, with $\delta_{\max} \triangleq \log \left(\frac{1}{1 - \varepsilon}\right)$. Note that a BF is not associated to a continuous probability density on $\Omega$ (e.g., it almost never holds that $\int_{\Omega} p_B(x)dx = 1$), but rather to a collection of discrete probability distributions, one for each point in $\Omega$ (thus, it always holds that $P(B_x = 0) + P(B_x = 1) = 1 \forall x \in \Omega$).

Due to the finite resolution of cameras and scanners, continuous BFs cannot be directly obtained as discrete BFs were obtained in Definition 3. In contrast, continuous BFs are obtained indirectly from discrete BFs (which are possibly obtained by one of the methods described in Definition 3). Let $\hat{B}$ be a discrete BF defined on a partition $\Pi(\hat{\Omega}) = \{\hat{\Omega}_1, \ldots, \hat{\Omega}_n\}$. Then, for each partition element $\hat{\Omega}_i$, and for each point $x \in \hat{\Omega}_i$, the success rate of the Bernoulli variable $B_x$ is defined as $p_B(x) \triangleq p_B(\hat{B})$. The BF $B$ produced in this fashion will be referred to as the BF produced by the discrete BF $\hat{B}$. Intuitively, $p_B$ extends $p_{\hat{B}}$ from every element of $\Pi(\hat{\Omega})$ to every point $x \in \Omega$. Note that this definition is analogous to the definition of a continuous shape produced from a discrete shape in Definition 4.
Let $\Omega' \subset \mathbb{R}^d$ be a set referred to as the canonical set, let $\Omega \subset \mathbb{R}^d$ be a second set referred to as the world set, and let $T : \Omega' \rightarrow \Omega$ be a bijective transformation between these sets. Given a BF $B$ in $\Omega'$ with success rates $p_B(x)$, the transformed BF $B_T$ in $\Omega$ is defined as $B_T \triangleq B \circ T^{-1}$ with success rates $p_{B_T}(x) \triangleq p_B(T^{-1}x)$.

**Definition 6 (Log-probability of a continuous shape)** Let $B$ be a BF in $\Omega$ with success rates given by the function $p_B(x)$, let $S$ be a continuous shape also in $\Omega$, and let $u_o > 0$ be a scalar called the equivalent unit size. We define the log-probability that a shape $S$ is produced by a BF $B$, by extension of the log-probability of discrete shapes in (11), as

$$\log P(B = S) \triangleq \frac{1}{u_o} \left[ Z_B + \int_{\Omega} S(x)\delta_B(x) \, dx \right],$$

(14)

where $Z_B$ and $\delta_B$ are respectively the constant term and the logit function of $B$.

Several things are worth noting in this definition. First, note that if there is a uniform partition $\Pi(\Omega) = \{\Omega_1, \ldots, \Omega_n\}$ with $|\Omega_i| = u_o \forall i$, and if the continuous shape $S$ and the BF $B$ are respectively produced by the discrete shape $\hat{S}$ and the discrete Bernoulli field $\hat{B}$ defined on $\Pi(\Omega)$, then $\log P(B = S) = \log P(\hat{B} = \hat{S})$.

For this reason we said that the definition in (14) extends the definition in (11). However, keep in mind that in the case of a continuous shape (14) is not a log-probability in the traditional sense, but rather it extends the definition to cases in which $S(x)$ is not produced from a discrete shape and $\delta_B(x)$ is not piecewise constant in a partition of $\Omega$.

Second, note that while (14) provides the “log-probability” density that a given continuous shape is produced by a BF, sampling from a BF is not guaranteed to produce a continuous shape (because the resulting set might not satisfy the regularity conditions in Definition 3). Nevertheless, this is not an obstacle since in this work we are only interested in computing log-probabilities of continuous shapes that are given, not on sampling from BF's.

Third, note in (14) that the log-probability of a continuous shape is the product of two factors: 1) the inverse of the unit size, which only depends on the partition (but not on the shape); and 2) a term (in brackets) that does not depend on the partition. In the case of continuous shapes, $u_o$ in the first factor is not the unit size of the partition (there is no partition defined in this case) but rather a scalar defining the unit size of an equivalent partition in which the range of log-probability values obtained would be comparable. The second factor is the sum of a constant term that only depends on the BF $B$, and a second term that also depends on the continuous shape $S$.

Fourth, the continuous shape representation in Definition 6 overcomes the limitations of the discrete representation pointed out above. More specifically, by considering continuous shapes, (14) can be computed even if a discrete shape and a discrete BF are defined on partitions that are not aligned, allowing us greater freedom in the choice on the transformations ($T$) that can be applied to the BF. Furthermore, the role of the partition is “decoupled” from the role of the BF and the shape, allowing us to compute (14) independently of the resolution of the partitions.

### 5.3 Semi-discrete shapes

As mentioned at the beginning of this section, discrete shapes will be used to obtain a lower bound for the log-probability of a continuous shape. Unfortunately, upper bounds for the log-probability derived using discrete shapes are not very tight. For this reason, to obtain upper bounds for this log-probability, we need to introduce the third shape representation, that of semidiscrete shapes.

**Definition 7 (Semidiscrete shapes)** Given a partition $\Pi(\Omega) = \{\Omega_1, \ldots, \Omega_n\}$ of the set $\Omega \subset \mathbb{R}^d$, the semidiscrete shape $\tilde{S}$ is defined as the function $\tilde{S} : \Pi(\Omega) \rightarrow [0, |\Omega|]$, that associates to each element $\Omega_i$ in the partition a real number in the interval $[0, |\Omega_i|]$, i.e., $\tilde{S}(\Omega_i) \in [0, |\Omega_i|]$ (Fig. 5c).

Given a continuous shape $S$ in $\Omega$, we say that this shape produces the semidiscrete shape $\tilde{S}$, denoted as $S \sim \tilde{S}$ or $\tilde{S} \sim S$, if $\tilde{S}(\Omega_i) = |S \cap \Omega_i|$ for $i = 1, \ldots, n$. An intuition that will be useful later to understand the derivation of the upper bounds is that a semidiscrete shape produced from a continuous shape “remembers” the measure of the continuous shape inside each element of the partition, but “forgets” where exactly the shape is located inside the element.

Given two continuous shapes in $\Omega$, $S_1$ and $S_2$, that produce the the same semidiscrete shape $\tilde{S}$ in the partition $\Pi(\Omega)$, we say the these continuous shapes are related (denoted as $S_1 \sim S_2$). The nature of this relationship is explored in the next proposition.

**Proposition 1 (Equivalent classes of continuous shapes)** The relationship $\sim$ defined above is an equivalence relation.

*Proof: The proof of this proposition is trivial from Definition 7.*
We will say that these continuous shapes are equivalent in the partition \( \Pi(\Omega) \), and by extension, we will also say that they are equivalent to \( \bar{S} \) (i.e., \( S_i \sim \bar{S} \)).

**Proposition 2 (Relationships between shape representations)** Let \( \Pi(\Omega) = \{\Omega_1, \ldots, \Omega_n\} \) be an arbitrary partition of a set \( \Omega \), and let \( \bar{\mathcal{S}}(\Pi) \) and \( \mathcal{S}(\Pi) \) be the sets of all discrete and semidiscrete shapes, respectively, defined on \( \Pi(\Omega) \). Let \( S \) be the set of all continuous shapes in \( \Omega \). Then,

\[
\begin{align*}
\{ & S : S \sim \bar{S}, \bar{S} \in \bar{\mathcal{S}}(\Pi) \} \subset \mathcal{S}, \quad \text{and}, \\
\{ & S : S \sim \bar{S}, \bar{S} \in \bar{\mathcal{S}}(\Pi) \} = \mathcal{S}.
\end{align*}
\]

(15) (16)

*Proof:* The proof of this proposition is trivial from definitions 3, 4, and 7. \( \square \)

### 5.4 LCDFs and summaries

So far we have introduced three different shape representations and established relationships among them. In this section we introduce the concepts of logit cumulative distribution functions (LCDFs) and summaries. These concepts will be necessary to use discrete and semidiscrete shapes to bound the log-probability of continuous shapes, and hence, to bound the evidence.

Intuitively, a LCDF “condenses” the “information” of a BF in a partition element into a monotonous function. A summary then further “condenses” this “information” into a single vector of fixed length. Importantly, the summary of a BF in a partition element can be used to bound the evidence \( L(\Pi) \) and can be computed in constant time, regardless of the number of pixels in the element.

After formally defining LCDFs and summaries below, we will prove in this Section some of the properties that will be needed in Section 6 to compute lower and upper bounds for \( L(\Pi) \). In the remainder of this section, unless stated otherwise, all partitions, shapes and BFs are defined on a set \( \Omega \subset \mathbb{R}^d \).

**Definition 8 (Logit cumulative distribution function)** Given the logit function \( \delta_B \) of a BF \( B \) and a partition \( \Pi \), the logit cumulative distribution function (or LCDF) of the BF \( B \) in \( \Pi \) is the collection of functions \( D_B = \{ D_{B,\omega} \}_{\omega \in \Pi} \), where each function \( D_{B,\omega} : [-\delta_{\max}, \delta_{\max}] \to [0, |\omega|] \) is defined as

\[
D_{B,\omega}(\delta) \triangleq |\{ x \in \omega : \delta_B(x) < \delta \}| \quad (\omega \in \Pi).
\]

(17)

It must be noted that this definition is consistent, since from Definition 3, the logit function is measurable. The LCDF is named by analogy to the probability cumulative distribution function, but must not be confused with it. Informally, a LCDF “condenses” the information of the BF by “remembering” the values taken by the logit function inside each partition element, but “forgetting” where those values are inside the element. This relationship between BFs and their LCDFs is analogous to the relationship between continuous and semidiscrete shapes.

Equation (17) defines a non-decreasing and possibly discontinuous function. To see that this function is non-decreasing, note that the set \( \chi_1 \triangleq \{ x \in \omega : \delta_B(x) < \delta_1 \} \) is included in the set \( \chi_2 \triangleq \{ x \in \omega : \delta_B(x) < \delta_2 \} \) if \( \delta_1 \leq \delta_2 \). Moreover, this function is not necessarily strictly increasing because it is possible to have \( D_{B,\omega}(\delta_1) = D_{B,\omega}(\delta_2) \) with \( \delta_1 < \delta_2 \) if the measure of the set \( \{ x \in \omega : \delta_1 \leq \delta_B(x) < \delta_2 \} \) is zero. To see that the function defined in (17) can be discontinuous, note that this function will have a discontinuity whenever the function \( \delta_B(x) \) is constant on a set of measure greater than zero.

Later in this section we will use the inverse of a LCDF, \( D_{B,\omega}^{-1}(a) \). If \( D_{B,\omega}(\delta) \) were strictly increasing and continuous, we could simply define \( D_{B,\omega}^{-1}(a) \) as the unique real number \( \delta \in [-\delta_{\max}, \delta_{\max}] \) such that \( D_{B,\omega}(\delta) = a \). For general LCDFs, however, this definition does not produce a value for every \( a \in [0, |\omega|] \). Instead we use the following definition.

**Definition 9 (Inverse LCDF)** The inverse LCDF \( D_{B,\omega}^{-1}(a) \) is defined as (see Fig. 6)

\[
D_{B,\omega}^{-1}(a) \triangleq \inf \{ \delta : D_{B,\omega}(\delta) \geq a \}.
\]

(18)

To avoid pathological cases, we will only consider in this work LCDFs whose inverse is continuous almost everywhere (note that this imposes an additional restriction on the logit function).
Definition 10 (X-axis crossing point) We define the x-axis crossing point, $A_{B,\omega}$, as

$$A_{B,\omega} \triangleq \frac{D_{B,\omega}(0^-) + D_{B,\omega}(0^+)}{2}.$$  \hspace{1cm} (19)

This quantity has the property that $D_{B,\omega}^{-1}(a) \geq 0$ if $a \geq A_{B,\omega}$, and $D_{B,\omega}^{-1}(a) \leq 0$ if $a \leq A_{B,\omega}$.

Definition 11 (Summaries) Given a partition $\Pi$, a summary of a LCDF $D_B = \{D_{B,\omega}\}_{\omega \in \Pi}$ in the partition is a functional that assigns to each function $D_{B,\omega}$ ($\omega \in \Pi$) in the LCDF a vector $Y_{B,\omega} \in \mathbb{R}^d$. The name “summary” is motivated by the fact that the “infinite dimensional” distribution is “summarized” by just $dY$ real numbers.

Two types of summaries are used in this article: m-summaries and mean-summaries. Given $m > 0$, an m-summary assigns to each function $D_{B,\omega}$ in the LCDF the $(2m+1)$-dimensional vector $Y_{B,\omega} = \left[ \hat{Y}_{B,\omega}^{-m} \ldots \hat{Y}_{B,\omega}^m \right]^T$ of equispaced samples of the LCDF, i.e.,

$$\hat{Y}^j_{B,\omega} \triangleq D_{B,\omega}\left(\frac{j\delta_{\max}}{m}\right) \quad (j = -m, \ldots, m).$$ \hspace{1cm} (20)

Note that since the LCDF is known to be a non-decreasing function, the information in the m-summary can be used to bound the inverse LCDF (Fig. 7). Specifically, we know that $D_{B,\omega}^{-1}(a) \leq \frac{2\omega(m)}{m}(j + 1)$ for $a \in \left[ \hat{Y}_{B,\omega}^{-j}, \hat{Y}_{B,\omega}^{j+1} \right]$, which can be written as

$$D_{B,\omega}^{-1}(a) \leq \frac{D_{Y_{B,\omega}}^{-1}(a)}{m} \triangleq \frac{\delta_{\max}}{m} J(a), \forall a \in [0, |\omega|],$$ \hspace{1cm} (21)

with

$$J(a) \triangleq \min \left\{ j : \hat{Y}_{B,\omega}^j \geq a \right\} = \left| \left\{ j : \hat{Y}_{B,\omega}^j < a \right\} \right| - m.$$ \hspace{1cm} (22)

These bounds will be used in turn to compute an upper bound for $L(H)$.

The second type of summaries, referred to as mean-summaries, will be used to compute a lower bound of $L(H)$. The mean-summaries assign to each function $D_{B,\omega}$ the scalar

$$\hat{Y}_{B,\omega} \triangleq \int_0^{|\omega|} D_{B,\omega}^{-1}(a) \, da = \int_{\omega} \delta_B(x) \, dx.$$ \hspace{1cm} (23)

The last equality follows by setting $\hat{S}(\omega) = |\omega|$ in (29) and is proved later in Lemma 1. Note that this equality provides the means to compute the mean-summary directly from the log function, without having to compute the LCDF.

One of the main properties of summaries is that, for certain kinds of sets, they can be computed in constant time regardless of the number of elements (e.g., pixels) in these sets. Next we show how to compute the mean-summary $\hat{Y}_{B,\phi}$ and the m-summaries $\hat{Y}_{B,\omega}$ of a BF $B$ for the set $\Phi \subset \Omega$ (defined below). For simplicity we assume that $\Omega \subset \mathbb{R}^2$, but the results presented here immediately generalize to higher dimensions. We also assume that $\Pi(\Omega) = \{\Omega_1, \Omega_2, \ldots, \Omega_{n,n}\}$ is a uniform partition of $\Omega$ organized in rows and columns, where each partition element $\Omega_{i,j}$ (in the i-th row and the j-th column) is a square of area $|\Omega_{i,j}| = u_0$. We assume that $B$, defined by its log function $\delta_B(x)$ ($x \in \Omega$), was obtained from a discrete shape prior $B$ in $\Pi(\Omega)$ (as described in Definition 5), and therefore $\delta_B(x) \triangleq \delta_B(i,j)$ $\forall x \in \Omega_{i,j}$. And finally, we assume that $\Phi$ is an axis-aligned rectangular region containing only whole pixels (i.e., not parts of pixels). That is,

$$\Phi \triangleq \bigcup_{i,j} \Omega_{i,j}. \hspace{1cm} (24)$$

In order to compute the mean-summary $\hat{Y}_{B,\phi}$, note that from (23),

$$\hat{Y}_{B,\phi} = \sum_{i,j} \int_{\Omega_{i,j}} \delta_B(x) \, dx = u_0 \sum_{i,j} \delta_B(i,j).$$ \hspace{1cm} (25)

The sum on the rhs of (25) can be computed in constant time by relying on integral images [11], an image representation precisely proposed to compute sums in rectangular domains in constant time. To accomplish
this, integral images precompute a matrix where each pixel stores the cumulative sum of the values in pixels with lower indices. The sum in \( \text{Eq. 25} \) is then computed as the sum of four of these precomputed cumulative sums.

The formula to compute the \( m \)-summary \( \tilde{Y}_{B,\Phi} \) is similarly derived. From \( \text{Eq. 20} \), and since \( \delta_B \) is constant inside each partition element, it holds for \( k = -m, \ldots, m \) that

\[
\tilde{Y}_{B,\Phi}^k = \left\{ \mathbf{x} \in \Phi : \delta_B(\mathbf{x}) < \frac{k\delta_{\max}}{m} \right\} = u_o \left\{ (i, j) : \right. \\
\left. i_L \leq i \leq i_U, j_L \leq j \leq j_U, \delta_B(i, j) < \frac{k\delta_{\max}}{m} \right\} \\
\text{(26)}
\]

Let us now define the matrices \( I_k \) \((k = -m, \ldots, m)\) as

\[
I_k(i, j) = \begin{cases} 
1, & \text{if } \delta_B(i, j) < \frac{k\delta_{\max}}{m} \\
0, & \text{otherwise.} 
\end{cases} \quad \text{(27)}
\]

Using this definition, \( \text{Eq. 20} \) can be rewritten as

\[
\tilde{Y}_{B,\Phi}^k = u_o \sum_{i_L \leq i \leq i_U, j_L \leq j \leq j_U} I_k(i, j), \quad \text{(28)}
\]

which as before can be computed in \( O(1) \) using integral images.

Before deriving formulas to bound \( L(H) \), we need the following two results.

**Proposition 3 (Equivalent classes of BFs)** Let \( \Pi \) be a partition, let \( B_1 \) and \( B_2 \) be two BFs, and consider the following binary relations: 1) \( B_1 \) and \( B_2 \) are related (denoted as \( B_1 \sim_B B_2 \)) if they produce the same LCDF \( D = \{D_x\}_{x \in \Pi} \); and 2) \( B_1 \) and \( B_2 \) are related (denoted as \( B_1 \sim_Y B_2 \)) if they produce the same summary \( Y = \{Y_x\}_{x \in \Pi} \). Then, “\( \sim_D \)” and “\( \sim_Y \)” are equivalence relations.

**Proof:** Immediate from definitions \( \text{8} \) and \( \text{11} \). \( \Box \)

In the first case (\( \sim_D \)) we will say that these BFs are equivalent in the partition \( \Pi \) with respect to distributions. Abusing the notation, we will also say that they are equivalent to (or compatible with) the LCDF (i.e., \( B_1 \sim D \)). Similarly, in the second case (\( \sim_Y \)) we will say that these BFs are equivalent in the partition \( \Pi \) with respect to summaries. Abusing the notation, we will also say that they are equivalent to (or compatible with) the summary (i.e., \( B_1 \sim Y \)). Note that if two BFs are equivalent with respect to a LCDF, they are also equivalent with respect to any summary of the LCDF. The reverse, however, is not necessarily true.

**Lemma 1 (Properties of LCDFs and \( m \)-Summaries)** Let \( \Pi \) be an arbitrary partition, and let \( \tilde{S} \) and \( D = \{D_x\}_{x \in \Pi} \) be respectively a semidiscrete shape and a LCDF in this partition. Then:

1. Given a BF \( B \) such that \( B \sim D \), it holds that (Fig. 7)

\[
sup_{\tilde{S} \sim \tilde{S}} \int_{\Omega} \delta_B(x)S(x) \, dx = \sum_{\omega \in \Pi} |\omega| D_{\omega}^{-1}(a) \, da. \quad \text{(29)}
\]

2. Similarly, given a continuous shape \( S \), such that \( S \sim \tilde{S} \), it holds that

\[
sup_{B \sim D} \int_{\Omega} \delta_B(x)S(x) \, dx = \sum_{\omega \in \Pi} |\omega| D_{\omega}^{-1}(a) \, da. \quad \text{(30)}
\]

3. Moreover, for any \( \alpha \in [0, |\omega|] \), the integrals on the rhs of \( \text{Eq. 29} \) and \( \text{Eq. 30} \) can be bounded as

\[
\int_0^{|\omega|} D_{\omega}^{-1}(a) \, da \leq \frac{\delta_{\max}}{m} \left[ J(\alpha) \left( \tilde{Y}_{\omega}(\alpha) - \alpha \right) + \sum_{j=J(\alpha)+1}^{m-1} \left( \tilde{Y}_{\omega}^{j-1} - \tilde{Y}_{\omega}^j \right) (j + 1) \right], \quad \text{(31)}
\]

where \( J(\alpha) \) is as defined in \( \text{Eq. 22} \).

4. The rhs of \( \text{Eq. 29} \) and \( \text{Eq. 30} \) are maximum when \( \tilde{S}(\omega) = |\omega| - A_\omega \), thus

\[
sup_{\tilde{S}} \sum_{\omega \in \Pi} |\omega| D_{\omega}^{-1}(a) \, da = \sum_{\omega \in \Pi} \int_{A_\omega} |\omega| D_{\omega}^{-1}(a) \, da. \quad \text{(32)}
\]

**Proof:** Due to space limitations this proof was included in the supplementary material. \( \Box \)

This concludes the presentation of the general mathematical framework to compute and bound probabilities of shapes. This framework is used in the next section to bound the evidence for the problem defined in Section 4, and also in \( \text{Eq. 32} \) to bound the evidence for a more complex problem.

**6 Bounding mechanism**

We are now ready to develop the BM for the problem defined in Section 4. This BM is the second part of the H&B algorithm proposed to solve this problem. To develop the BM we proceed in three steps: first we derive an expression for the evidence \( L(H) \) that the FoAM will maximize (Section 6.1); second we show how to compute lower and upper bounds for the evidence for
a given partition (Section 6.2); and third, we show how to construct, incrementally, a sequence of increasingly finer partitions that will result in increasingly tighter bounds (Section 6.3).

6.1 Definition of the Evidence $L(H)$

In order to define the evidence $L(H)$ for the problem, we re-label the pixels of the image in (7) (denoted previously by $x$) as the elements of a uniform partition $H$, and separate the factors according to its state $q(\omega)$. Hence (7) is equal to

$$P(H) \prod_{\omega \in H} P_\omega(f(\omega)|q(\omega)) P(q(\omega)|H) = P(H) \prod_{\omega \in H} \left[ P_\omega(f(\omega)|q(\omega) = 0) P(q(\omega) = 0|H) \right]^{1-q(\omega)} \times \left[ P_\omega(f(\omega)|q(\omega) = 1) P(q(\omega) = 1|H) \right]^{q(\omega)}.$$  \hspace{1cm} (33)

Defining a BF $B_f$ based on the features observed in the input image, with success rates given by

$$p_{B_f}(\omega) \triangleq \frac{P_\omega(f(\omega)|q(\omega) = 1)}{P_\omega(f(\omega)|q(\omega) = 0) + P_\omega(f(\omega)|q(\omega) = 1)},$$  \hspace{1cm} (34)

assuming that all hypotheses are equally likely, and dividing (33) by the constant (and known) term

$$P(H) \prod_{\omega \in H} (P_\omega(f(\omega)|q(\omega) = 0) + P_\omega(f(\omega)|q(\omega) = 1)),$$  \hspace{1cm} (35)

we obtain an expression equivalent to (7),

$$\prod_{\omega \in H} \left[ (1 - p_{B_f}(\omega))(1 - p_{B_{1-H}}(\omega)) \right]^{1-q(\omega)} \times \left[ p_{B_f}(\omega) p_{B_{1-H}}(\omega) \right]^{q(\omega)}.$$  \hspace{1cm} (36)

(recall from Section 4 that $B_{1-H}$ is a BF with success rates $p_{B_{1-H}} \triangleq P(q(\omega) = 1|H)$). This expression can be further simplified by taking logarithms and using the variables introduced in Definition 3 to yield

$$Z_{B_f} + Z_{B_{1-H}} + \sum_{\omega \in \Omega} q(\omega) (\delta_{B_f}(\omega) + \delta_{B_{1-H}}(\omega)).$$  \hspace{1cm} (37)

Using the extension to the continuous domain explained in Definition 4 and substituting (37) into (9), (9) can be rewritten as

$$L'(H) \triangleq \sup_{q} \frac{1}{u_o} \left[ Z_{B_f} + Z_{B_{1-H}} + \int_{\Omega} q(x) (\delta_{B_f}(x) + \delta_{B_{1-H}}(x)) \, dx \right].$$  \hspace{1cm} (38)

Now, since $Z_{B_f}$ and $u_o$ are constant for every hypothesis and every continuous shape $q$, maximizing this expression is equivalent to maximizing

$$L(H) \triangleq Z_{B_{1-H}} + \sup_{q} \int_{\Omega} q(x) (\delta_{B_f}(x) + \delta_{B_{1-H}}(x)) \, dx.$$  \hspace{1cm} (39)

This is the final expression for the evidence. Due to the very large number of pixels in a typical image produced by a modern camera, computing $L(H)$ directly as the integral in (39) would be prohibitively expensive. For this reason, the next step is to derive bounds for (39) that are cheaper to compute than (39) itself, and are sufficient to discard most hypotheses. These bounds are derived in the next section.

6.2 Derivation of the bounds

In the following two theorems we derive bounds for the evidence of a hypothesis that can be computed from summaries of the BFs $B_f$ and $B_{1-H}$ (in (39)), instead of computing them from the BFs directly. Because summaries can be computed in $O(1)$ for each element in a partition, bounds for a given partition can be computed in $O(n)$ (where $n$ is the number of elements in the partition), regardless of the actual number of pixels in the image.

**Theorem 1 (Lower bound for $L(H)$)** Let $H$ be a partition, and let $\tilde{Y}_f = \{\tilde{Y}_{f,\omega}\}_{\omega \in \Omega}$ and $\tilde{Y}_{1-H} = \{\tilde{Y}_{1-H,\omega}\}_{\omega \in \Omega}$ be the mean-summaries of two unknown BFs in $H$. Then, for any $B_f \sim \tilde{Y}_f$ and any $B_{1-H} \sim \tilde{Y}_{1-H}$, it holds that $L(H) \geq L_H(H)$, where

$$L_H(H) \triangleq Z_{B_{1-H}} + \sum_{\omega \in \Omega} \mathcal{L}_{\omega}(H),$$  \hspace{1cm} (40)

$$\mathcal{L}_{\omega}(H) \triangleq \left( \tilde{Y}_{f,\omega} + \tilde{Y}_{1-H,\omega} \right) \hat{q}_\omega(\omega),$$  \hspace{1cm} (41)

and $\hat{q}_\omega$ is a discrete shape in $\Omega$ defined as

$$\hat{q}_\omega(\omega) \triangleq \begin{cases} 1, & \text{if } \left( \tilde{Y}_{f,\omega} + \tilde{Y}_{1-H,\omega} \right) > 0 \\ 0, & \text{otherwise}. \end{cases}$$  \hspace{1cm} (42)

**Proof:** Due to space limitations this proof was included in the supplementary material. □

**Theorem 2 (Upper bound for $L(H)$)** Let $H$ be a partition, and let $\tilde{Y}_f = \{\tilde{Y}_{f,\omega}\}_{\omega \in \Omega}$ and $\tilde{Y}_{1-H} = \{\tilde{Y}_{1-H,\omega}\}_{\omega \in \Omega}$ be the m-summaries of two unknown BFs in $H$. Let $\tilde{Y}_{f \oplus 1-H,\omega}$ ($\omega \in H$) be a vector of length $4m + 2$ obtained
by sorting the values in $\tilde{Y}_{f,\omega}$ and $\tilde{Y}_{H,\omega}$ (in ascending order), keeping repeated values, i.e.,

$$\tilde{Y}_{f,\omega} \oplus H,\omega \triangleq \left[ \tilde{Y}_{f,\omega}^{1} \oplus H,\omega, \ldots, \tilde{Y}_{f,\omega}^{4m+2} \oplus H,\omega \right]$$

$$\triangleq \text{SortAscending} \left( \tilde{Y}_{f,\omega} \cup \tilde{Y}_{H,\omega} \right).$$  \hspace{1cm} (43)

Then, for any $B_{f} \sim \tilde{Y}_{f}$ and any $B_{H} \sim \tilde{Y}_{H}$, it holds that $L(H) \leq L_{H}(\Pi)$, where

$$\overline{L}_{H}(\Pi) \triangleq Z_{B_{H}} + \sum_{\omega \in \Pi} \overline{L}_{\omega}(H),$$

$$\overline{L}_{\omega}(H) \triangleq \frac{\delta_{\max}}{m} \sum_{j=2m+1}^{4m+1} (j-2m) \left( \tilde{Y}_{f,\omega}^{j+1} \oplus H,\omega - \tilde{Y}_{f,\omega}^{j} \oplus H,\omega \right).$$  \hspace{1cm} (44)

It also follows that the continuous shape that maximizes (39) is equivalent to a semidiscrete shape $\tilde{q}_{*}$ in $\Pi$ that satisfies

$$\tilde{q}_{*}(\omega) \in \left[ \omega - \tilde{Y}_{f,\omega}^{2m+1} \oplus H,\omega, \omega - \tilde{Y}_{f,\omega}^{2m} \oplus H,\omega \right] \forall \omega \in \Pi.$$  \hspace{1cm} (46)

Proof: Due to space limitations this proof was included in the supplementary material. \hfill \Box

Theorems 1 and 2 presented formulas to compute lower and upper bounds for $L(H)$, respectively, for a given partition $\Pi$. Importantantly, these theorems also include formulas to compute a discrete shape $q_{*}$ and a semidiscrete shape $\tilde{q}_{*}$ that approximate (in the partition $\Pi$) the continuous shape $q$ that solves (39). In the next section we show how to reuse the computation spent to compute the bounds for a partition $\Pi_{k}$, to compute the bounds for a finer partition $\Pi_{k+1}$.

6.3 Incremental refinement of bounds

Given a partition $\Pi_{k}$ containing $h_{k}$ elements, it can be seen in [40] and [41] that the bounds for the evidence corresponding to this partition can be computed in $O(h_{k})$. In Section 3, however, we requested that the BM be able to compute these bounds in $O(1)$. In order to compute a sequence of progressively tighter bounds for a hypothesis $H$, where each bound is computed in $O(1)$, we inductively construct a sequence of progressively finer partitions of $\Omega$ for the hypothesis.

Let us denote by $\Pi_{k}(H) \triangleq \left\{ \Omega_{H,1}, \ldots, \Omega_{H,h_{k}} \right\}$ the $k$-th partition in the sequence corresponding to $H$. Each sequence is defined inductively by

$$\Pi_{1}(H) \triangleq \left\{ \Omega \right\},$$

$$\Pi_{k+1}(H) \triangleq \left[ \Pi_{k}(H) \setminus \omega_{k} \right] \cup \pi(\omega_{k}), \hspace{1cm} (k > 0).$$  \hspace{1cm} (47)

where $\omega_{k} \in \Pi_{k}(H)$ and $\pi(\omega_{k})$ is a partition of $\omega_{k}$. For each partition $\Pi_{k}(H)$ in the sequence, lower ($L_{k}(H)$) and upper ($L_{\omega_{k}}(H)$) bounds for the evidence $L(H)$ could be computed in $O(k)$ using [10] and [11], respectively. However, these bounds can be computed more efficiently by exploiting the form of (48), as

$$L_{k+1}(H) = L_{k}(H) - L_{\omega_{k}}(H) + \sum_{\omega \in \pi(\omega_{k})} L_{\omega}(H).$$  \hspace{1cm} (49)

(A similar expression for the upper bound $L_{k+1}(H)$ can be derived.) If the partition of $\omega_{k}$, $\pi(\omega_{k})$, is chosen to always contain a fixed number of sets (e.g., $|\pi(\omega_{k})| = 4 \forall k > 1$), then it can be seen in [19] that $O(1)$ evaluations of (11) are required to compute $L_{k+1}(H)$.

While any choice of $\omega_{k}$ from $\Pi_{k}(H)$ in [18] would result in a new partition $\Pi_{k+1}(H)$ that is finer than $\Pi_{k}(H)$, it is natural to choose $\omega_{k}$ to be the set in $\Pi_{k}(H)$ with the greatest local margin ($|L_{\omega_{k}}(H) - L_{\omega_{k}}(H)|$) since this is the set responsible for the largest contribution to the total margin of the hypothesis ($|L_{\omega_{k}}(H) - L_{\omega_{k}}(H)|$). In order to efficiently find the set $\omega_{k}$ with the greatest local margin, we store the elements of a partition in a priority queue, using their local margin as the priority. Hence to compute $\Pi_{k+1}(H)$ from $\Pi_{k}(H)$ (in [18]), we need to extract the element $\omega_{k}$ of the queue with the greatest local margin, and then insert each element in $\pi(\omega_{k})$ into the queue. Taken together these steps have, depending on the implementation of the queue, complexity of at least $O(\log h_{k})$ (where $h_{k}$ is the number of elements in the partition) [23]. In our case, however, it is not essential to process the elements in strictly descending margin order. Any element with a margin close enough to the maximum margin would produce similar results. Moreover, in our case we know that the margins belong to the interval $\left[ 0, L_{\Omega}(H) - L_{\Omega}(H) \right]$ and that they tend to decrease with time.

Based on these considerations we propose a queue implementation based on the untidy priority queue of Yatziv et al. [44], in which the operations GetMax and Insert both have complexity $O(1)$. This implementation consists of an array of buckets (i.e., singly-linked lists), where each bucket contains the elements whose margin is in an interval $I_{j}$. Specifically, suppose that the minimum and maximum margin of any element are known to be $M_{1}$ and $M_{1}$, respectively, and that $\rho > 1$ is a constant (we chose $\rho = 1.2$). The intervals are then defined to be $I_{j} \triangleq \left( [M_{1}\rho^{j-1}, M_{1}\rho^{j}] \right) (j = 1, \ldots, \lfloor \log_{\rho} M_{1}/M_{1} \rfloor)$, where $\lfloor \cdot \rfloor$ is the ceiling function. To speed up the GetMax operation, a variable $j_{\text{max}}$ keeps the index of the non-empty bucket containing the element with the largest margin. In the Insert operation, we simply compute the index $j$ of the corresponding bucket, insert the element in this bucket, and up-
date $j_{\text{max}}$ if $j > j_{\text{max}}$. In the \texttt{GetMax} operation, we return any element from the $j_{\text{max}}$-th bucket, and update $j_{\text{max}}$. Note that the margin of the returned element is not necessarily the maximum margin in the queue, but it is at least $1/\rho$ times this value. Since both operations (\texttt{Insert} and \texttt{GetMax}) can be carried out in $O(1)$, we have proved that (10) can also be computed in $O(1)$.

Moreover, since the bounds in (11) and (15) are tighter if the regions involved are close to uniform (because in this case, given the summary, there is no uncertainty regarding the value of any point in the region), this choice of $\omega_k$ automatically drives the algorithm to focus on the edges of the image and the prior, avoiding the need to subdivide and work on large uniform regions of the image or the prior.

This concludes the derivation of the bounds to be used to solve our problem of interest. In the next section we show results obtained using these bounds integrated with the FoAM described in Section 3.

7 Experimental results

In this section we apply the framework described in previous sections to the problem of simultaneously estimating the class, pose, and a denoised version (a segmentation) of a shape in an image. We start by analyzing the characteristics of the proposed algorithm on synthetic experiments (Section 7.1), and then present experiments on real data (Section 7.2). These experiments were designed to test and illustrate the proposed theory only. Achieving state-of-the-art results for each of the specific sub-problems would require further extensions of this theory.

7.1 Synthetic experiments

In this section we present a series of synthetic experiments to expose the characteristics of the proposed approach.

\textit{Experiment 1.} We start with a simple experiment where both the input image (Fig. 8a) and the shape prior are constructed from a single shape (the letter ‘A’). Since we consider a single shape prior, we do not need to estimate the class of the shape in this case, only its pose. In this situation the success rates $p_{B_f}$ of the BF corresponding to this image (Fig. 8a), and the success rates $p_{B_R}$ of the BF corresponding to the shape prior, are related by a translation $t$ (i.e., $p_{B_f}(x) = p_{B_R}(x - t)$). This translation is the pose that we want to estimate. In order to estimate it, we define four hypotheses and use the proposed approach to select the hypothesis $H$ that maximizes the evidence $L(H)$. Each hypothesis is obtained for a different translation (Fig. 8b), but for the same shape prior.

As described in Section 3, at the beginning of the algorithm the bounds of all hypotheses are initialized, and then during each iteration of the algorithm one hypothesis is selected, and its bounds are refined (Fig. 9). It can be seen in Fig. 9 that the FoAM allocates more computational cycles to refine the bounds of the best hypothesis (in red), and less cycles to the other hypotheses. In particular, two hypotheses (in cyan and blue) are discarded after spending just one cycle (the initialization cycle) on each of them. Consequently, it
can be seen in Fig. 10 that the final partition for the red hypothesis is finer than those for the other hypotheses. It can also be seen in the figure that the partitions are preferentially refined around the edges of the image or the prior (remember from (39) that image and prior play a symmetric role), because the partition elements around these areas have greater margins. In other words, the FoAM algorithm is “paying attention” to the edges, a sensible thing to do. Furthermore, this behavior was not intentionally “coded” into the algorithm, but rather it emerged as the BM greedily minimizes the margin.

To select the best hypothesis in this experiment, the functions to compute the lower and upper bounds (i.e., those that implement (41)-(42) and (45)-(46)) were called a total of 88 times each. In other words, 22 pairs of those that implement (41)-(42) and (45)-(46)) were called for reference only.

In other words, the amount of computation needed to solve a particular problem using the proposed approach only depends on the problem, not on the resolution of the input image.

Experiment 2. The next experiment is identical to the previous one, except that one hypothesis is defined for every possible integer translation that yields a hypothesis whose support is contained within the input image. This results in a total of 148,225 hypotheses. In this case, the set $A$ of active hypotheses when termination conditions were reached contained 3 hypotheses. We refer to this set as the set of solutions, and to each hypothesis in this set as a solution. Note that having reached termination conditions with a set of solutions having more than one hypothesis (i.e., solution) implies that all the hypotheses in this set have been completely refined (i.e., either $p_B$ or $p_B^H$ are uniform in all their partition elements).

To characterize the set of solutions $A$, we define the translation bias, $\mu_t$, and the translation standard deviation, $\sigma_t$, as

$$\mu_t \triangleq \left\| \frac{1}{|H|} \sum_{i=1}^{n} t_i - t_T \right\|,$$

and

$$\sigma_t \triangleq \sqrt{\frac{1}{|H|} \sum_{i=1}^{n} ||t_i - t_T||^2},$$

respectively, where $t_i$ is the translation corresponding to the $i$-th hypothesis in the set $A$ and $t_T$ is the true translation. In this particular experiment we obtained $\mu_t = 0$ and $\sigma_t = 0.82$, and the set $A$ consisted on the true hypothesis and the two hypotheses that are one pixel translated to the left and right. These 3 hypotheses are indistinguishable under the conditions of the experiment. There are two facts contributing to the uncertainty that makes these hypotheses indistinguishable: 1) the fact that the edges of the shape in the image and the prior are not sharp (i.e., not having probabilities of 0/1, see inset in Fig. 3i); and 2) the fact that $m < \infty$ in the $m$-summaries and hence some “information” of the LCDF is “lost” in the summary, making the bounds looser.

Figure 11 shows the evolution of the set $A$ of active hypotheses as the bounds get refined. Observe in this figure that during the refinement stage of the algorithm ($cycles > 0$ in the figure), the number of active hypotheses ($|A|$), the bias $\mu_t$, and the standard deviation $\sigma_t$ sharply decrease. It is interesting to note that during the first half of the initialization stage, because hypotheses are not discarded symmetrically around the true hypothesis, the bias increases.
**Experiment 3.** In the next experiment the hypothesis space is enlarged by considering not only integer translations, but also scalings. These scalings changed the horizontal and vertical size of the prior (independently in each direction) by multiples of 2%. In other terms, the transformations used were of the form

\[
T(x) = \text{Diag}(s)x + t,
\]

where \( t \triangleq [t_x, t_y] \) is an integer translation (i.e., \( t_x, t_y \in \mathbb{Z} \)), \( \text{Diag}(s) \) is a diagonal matrix containing the vector of scaling factors \( s \triangleq [s_x, s_y] \) in its diagonal, and \( s_x, s_y \in \{0.96, 0.98, 1, 1.02, 1.04\} \).

To characterize the set of solutions \( \mathcal{A} \) of this experiment, in addition to the quantities \( \mu_k \) and \( \sigma_k \) that were defined before, we also define the scaling bias, \( \mu_s \), and the scaling standard deviation, \( \sigma_s \), as

\[
\mu_s \triangleq \left| \frac{1}{|A|} \sum_{i=1}^{|A|} s_i - 1 \right|, \quad \text{and}
\]

\[
\sigma_s \triangleq \sqrt{\frac{1}{|A|} \sum_{i=1}^{|A|} \|s_i - 1\|^2},
\]

respectively, where \( s_i \) are the scaling factors corresponding to the \( i \)-th hypothesis in the set \( \mathcal{A} \) (the true scaling factors, for simplicity, are assumed to be \( s_x = s_y = 1 \)).

In this experiment the set of solutions consisted of the same three hypotheses found in Experiment 2, plus two more hypotheses that had a ±2% scaling error. The performance of the framework in this case, on the other hand, improved with respect to the previous experiment (note the \( \tau \) in Table 2). These results (summarized in Table 2) suggest that in addition to the translation part of the pose, the scale can also be estimated very accurately and efficiently.

**Table 2** Errors in the estimation of the position and scale of a known shape ‘A’ in a noiseless input image.

| \( |A| \) | \( \mu_k \) (pixels) | \( \sigma_k \) (pixels) | \( \mu_s \) (%) | \( \sigma_s \) (%) | \( \tau \) |
|---|---|---|---|---|---|
| 5 | 0 | 0.633 | 0 | 1.26 | 1.7 |

**Experiment 4.** The performance of the framework is obviously affected when the input image is corrupted by noise. To understand how it is affected, we run the proposed approach with the same hypothesis space defined in Experiment 2, but on images degraded by different kinds of noise (for simplicity we add the noise directly to the BF corresponding to the input image, rather than to the input image itself). Three kinds of noise have been considered (Fig. 13a): (1) additive, zero mean, white Gaussian noise with standard deviation \( \sigma \), denoted by \( \mathcal{N}(0, \sigma^2) \); salt and pepper noise, \( \text{SP}(P) \), produced by transforming, with probability \( P \), the success rate \( p(x) \) of a pixel \( x \) into \( 1 - p(x) \); and structured noise, \( S(\ell) \), produced by transforming the success rate \( p(x) \) into \( 1 - p(x) \) for each pixel \( x \) in rows and columns that are multiples of \( \ell \). When adding Gaussian noise to a BF some values end up outside the interval \([0, 1]\). In such cases we trim these values to the corresponding extreme of the interval. The results of these experiments are summarized in Table 3.
Table 3 Effect of corrupting the BF corresponding to the input image with noise.

| Noise    | |A| | μ_k (pixels) | σ_k (pixels) | τ |
|----------|---|---|---------------|---------------|---|
| N(0,0.15^2) | 12 | 0.16 | 1.41 | 4.59 |
| N(0,0.30^2) | 11 | 0 | 1.34 | 2.66 |
| N(0,0.45^2) | 11 | 0 | 1.34 | 3.12 |
| S(P(0.083) | 3 | 0 | 0.81 | 1.6 |
| S(P(0.125) | 6 | 0.5 | 1.08 | 2.76 |
| S(P(0.25) | 3 | 0 | 0.81 | 6.28 |
| S(8) | 3 | 0.47 | 1.15 | 42.77 |

By comparing tables 1 and 2, we observe that the approach is relatively immune to the levels and kinds of noise applied, since μ_k and σ_k did not increase much. However, the effects of the different kinds of noise are different. Adding moderate amounts of Gaussian noise, by making it uncertain whether each pixel in the image belongs to the foreground or background, increased by more than 3 times the size of the set of solutions, while doubling the amount of computation required (compare τ in tables 1 and 3). In contrast, moderate amounts of salt and pepper noise almost did not affect the set of solutions found, but they dramatically increased the necessary computation (because this kind of noise created many more “edges” in the image). To understand the effect of the position of the errors introduced by the noise, we adjusted the level of the structured noise ℓ so that the same (approximate) number of pixels were affected by it and by the salt and pepper noise (i.e., approximately the same number of pixels were affected by S(P(0.083)) and S(24), by S(P(0.125)) and S(16), and by S(P(0.25)) and S(8)). Because the structured noise concentrates the “errors” in some parts of the image, this kind of noise increased slightly the size of the set of solutions, but it did not have a consistent effect on the errors or the amount of computation required.

Fig. 13 (a) BFs corresponding to the input image after being degraded by three different kinds of noise (from left to right): Gaussian noise N(0,0.45^2); Salt and pepper noise S(P(0.25)); and structured noise S(8). (b-c) Discrete shape  # (b), and semidiscrete shape  # (c), estimated for the best solution after 50 refinement cycles. Each shape was obtained for the corresponding input image depicted in the same column in (a). (d-e) Discrete (d) and semidiscrete (e) shapes obtained when termination conditions were achieved, for each of the input images in (a). The colors of the semidiscrete shapes at each partition element ω represent the fraction of the element (i.e.,  #(ω)/|ω|) that is covered by the shape (use the colorbar in Fig. 8 to interpret these colors).

Experiment 5. Another factor that affects the framework’s performance is the “shape” of the prior. As mentioned before (in Section 6.3), the bounds computed for a partition element are tightest when either one of the BFs corresponding to the prior or the image are uniform in the element. This is because when one of the BFs is uniform, given the m-summary, there is no uncertainty about the location of the values of δ(x) in the element. Due to this fact, shapes that are uniform on elements of a coarser partition (referred below as “coarser” shapes) are processed faster than shapes that are uniform only on the elements of a finer partition (referred below as “finer” shapes). This statement is quantified in Table 4 which summarizes the results of five experiments.

Each of these experiments is identical to Experiment 2, except that a different shape was used instead of the shape ‘A’. These shapes are depicted on the first column of Table 4. Note that changing the shape in these experiments did not affect the set of solutions found (in
Table 4 Effect of the shape on the framework’s performance.

| Shape | | $|A|$ | $\mu_t$ (pixels) | $\sigma_t$ (pixels) | $\tau$ |
|-------|---|--------|------------------|------------------|--------|
| □     | 1 | 1      | 0                | 0                | 1.00   |
| ◆     | 1 | 1      | 0                | 0                | 1.04   |
| ▲     | 1 | 0      | 0                | 1.10             |        |
| ▼     | 1 | 0      | 0                | 1.50             |        |
| ★     | 1 | 0      | 0                | 1.77             |        |

all cases this set contains only the true solution and thus $\mu_t = \sigma_t = 0$, but it did affect the amount of computation required in each case (see the column labeled ‘$\tau$’). In particular, note that since the shapes are roughly sorted from “coarser” to “finer” (as defined above), the total computation required correspondingly increases. Interestingly, to estimate the pose of the first shape (a square of size 128 × 128), only one pair of bounds were computed (in constant time) per hypothesis (instead of processing the 16,384 pixels of the shape prior).

7.2 Experiments on real data

In this section we apply the proposed framework to the problem of estimating the class, pose, and segmentation of a shape (a letter in this case), for each image in a set of testing images. In order to do this, as before, we define multiple hypotheses and use the proposed framework to find the group of hypotheses that best explains each test image.

More specifically, one hypothesis is defined for each transformation (described by (52)) and for each class. To save memory, and since it was shown before that hypotheses that are far from the ground truth are easily discarded, we only considered integer translations (around the ground truth) of up to 5 pixels in each direction (i.e., $t_x, t_y \in \{-5, -4, \ldots, 5\}$), and scalings (around the ground truth) by multiples of 4% (i.e., $s_x, s_y \in \{0.92, 0.96, \ldots, 1.08\}$). In contrast with the experiments presented in the previous section where only one class was considered (and hence there was no need to estimate it), in this section multiple classes are considered and estimating the class of the shape is part of the problem. We consider 6 different priors for each letter of the English alphabet, giving a total of $26 \times 6 = 156$ different priors (and classes). This results in an initial set of hypotheses containing 471,900 hypotheses (121 translations × 25 scalings × 156 priors).

To construct the priors for each letter, we compiled a set of training shapes from the standard set of fonts in the Windows operating system. This set was pruned by discarding italicized fonts and other fonts that were considered outliers (e.g., wingdings). The remaining shapes were then divided into six subsets using k-medoids clustering [12], where the distance $d(S_1, S_2)$ between two shapes $S_1$ and $S_2$ was defined as the number of different pixels between the two shapes (i.e., $d(S_1, S_2) \triangleq |S_1 \cup S_2 \setminus S_1 \cap S_2|$). Then all the letters in each cluster were aligned with respect to the medoid of the cluster by finding the optimal translation and scaling that minimizes the distance $d$ mentioned above, and (13) was used to compute the prior corresponding to the cluster. The six priors corresponding to the letters ‘A’ and ‘T’ are shown in Fig. 14. The number of priors per class (six) was chosen to maximize the classification performance $P_1$ (defined below). We also tested having a variable number of clusters per class, but doing so did not improve the classification performance. The resulting priors used in the following experiments, as well as the set of training shapes used to construct them, can be downloaded from [30].

The testing images, on the other hand, were obtained by extracting rectangular regions (containing each single letter) from old texts exhibiting decoloration and ink bleedthough, among other types of noise (see examples in Fig. 15 and in the images of these old texts in [30]). The corresponding BF s were obtained from these images by background subtraction using (12). The probability density functions (pdf’s) $p(f|q = 0)$ and $p(f|q = 1)$ in that equation were learned from sets of pixels in the background and foreground that were manually selected for each image. In this case we learned a single pdf for the background, and a single pdf for the foreground (not a different pdf for each pixel). While more sophisticated methods could have been used to obtain these BF s, this is not the goal of this article, and in fact, noisy BF s are desirable to test the robustness of the approach. This testing set, containing 218 letters, is used to measure the classification performance of the proposed framework. It is important to note that the testing and training sets come from different sources (i.e., there is no overlap between the testing and training sets).

Since the proposed approach does not necessarily associate a single class to each testing image (because the set of solutions obtained for an image might contain solutions of different classes), we report the performance of the framework with a slight modification of...
but considering only the solutions whose upper bound define the confusion matrix $C$. For this purpose we know what the classification performance is when only for all testing images of class $i$ solutions of class $j$ are considered, and that $\gamma$ is defined as in Section 3, and hence all the solutions $\gamma$ are the maximum lower and upper bounds, respectively, of any solution. Note that $\gamma_0 = \gamma = L$ when $\gamma$ is defined as in Section 3 and hence all the solutions are considered, and that $\gamma_1 = \bar{L}$ and hence only the solution with the largest upper bound is considered. The confusion matrices $C_{0.5}$ and $C_1$ are also shown in Fig. 16.

Similarly, the total classification performance, $P_3$, is defined as the number of solutions of the correct class (accumulated over all the images in the testing set), divided by the total number of solutions. As above, the solutions considered are only those whose upper bound is greater or equal than $\gamma_\beta$. The total performance obtained in this case was $P_0 = 82.5\%$, $P_{0.5} = 86.5\%$ and $P_1 = 90.4\%$. As expected, when only the best solutions are considered, the performance improves (i.e., $P_1 \geq P_{0.5} \geq P_0$).

For completeness, Table 5 summarizes the average pose estimation “errors” obtained in this case. The quantities $|A_\beta|$, $P$, $\bar{P}$, and $\bar{P}_S$ in this table were respectively obtained as the average of the quantities $|A_\beta|$, $\mu_4$, $\sigma_4$, $\mu_5$, $\sigma_5$, and $\sigma_S$ (defined before) over all images in the testing set. In contrast with the synthetic experiments reported in Section 7.4 for the experiments reported in this section the ground truth pose is not available.

For this reason, and for lack of a universally trusted definition of the true pose (which is ill defined when the shapes to be aligned are different and unknown), in these experiments the ground truth pose was defined by the center (for the location) and the size (for the scale) of a manually defined bounding box around each letter. Therefore, the errors in Table 5 must be analyzed with caution, since it is not clear that this arbitrary definition of the ground truth should be preferred over the solutions returned by the framework. In fact, in all the inspected cases (e.g., see Fig. 18), we did not observe a clear “misalignment.”

Table 5 Mean “errors” in the estimation of the position and scale of the unknown shapes in the images of the testing set. The mean is computed over all the 218 images in this set.

| $|A_\beta|$ | $P$ (pixels) | $\bar{P}$ (pixels) | $\bar{P}_S$ (%) | $\bar{P}_S$ (%) |
|---|---|---|---|---|
| 271 | 1.36 | 2.26 | 2.70 | 6.64 |

Note in Table 5 that the average number of solutions per image (271) is only a small fraction (0.06%) of the total number of hypotheses (471,900). Moreover, these solutions are (in general) concentrated near the “ground truth” defined above (judging from the mean “errors” in the table).

To illustrate the types of segmentations obtained by the framework, Fig. 17 shows the shapes estimated for each of the BF’s in (a). As in Fig. 13 the colors in these semisemidiscet shapes represent the fraction of each partition element that is “covered” by the shape.
Fig. 18 Representative examples of the solutions found by
the proposed framework, given by their corresponding BFs.
Solutions in the top row are of the correct class, while solu-
tions in the bottom row are of a different class. These compos-
itive images are produced by overlaying the BF corresponding
to the testing image (in the blue channel) and the BF corre-
sponding to the solution (in the red channel).

bottom row a solution from a different class is depicted.
It can be seen in the figure that when the fit of the cor-
rect solution is not very good, solutions of an incorrect
class cannot be discarded. This happens in the case of
rotated or distorted letters (e.g., ‘D’ and ‘N’ in the fig-
ure) and in cases where there is not a good fit in the
database of priors (e.g., ‘F’ in the figure). These erros
possibly can be overcome by considering richer trans-
formations and/or more numerous priors, respectively
(more on this in the next section).

8 Conclusions

This article presented a new type of algorithms, namely
hypothesize-and-bound algorithms, to significantly trim
down the amount of computation required to select the
hypothesis that best explains an input (e.g., an im-
age), from a set of predefined hypotheses. These algo-
rithms are specifically targeted to perceptual problems
in which the large size of the input imposes that these
problems be solved by “paying attention” to a small
fraction of the input only.

These algorithms have two important properties: 1)
they are guaranteed to find the set of optimal hypothe-
ses; and 2) the total amount of computation required
by them is decoupled from the “resolution” of the input,
and only depends on the current problem.

After describing the general paradigm of H&B algo-
rithms, we instantiated this paradigm to the problem
of simultaneously estimating the class, pose, and a de-
noised version of a 2D shape in a 2D image, by incorpo-
rating prior knowledge about the possible shapes that
can be found in the image. In order to instantiate the
paradigm, we developed a novel theory of shapes and
shape priors that allowed us to develop the BM for this
particular problem.

We consider that the theory and algorithms pro-
posed here are not just limited to solve the problem de-
scribed in this article, but rather that they are general
enough as to find application in many other domains.

In fact, the application of the exact same FoAM (de-
scribed in Section 3) and theory of shapes (described in
Section 5) have already been instantiated to solve the
problem of simultaneously estimating the class, pose,
and a 3D reconstruction of an object from a single 2D
image [32].

Even though the contributions of this work are mainly
theoretical, we showed preliminary results of the pro-
posed framework on a real problem. Nevertheless, we
acknowledge that to provide state-of-the-art results the
framework needs to be extended in several directions
(which is beyond the scope of the current article). For
example, a limitation of the current framework is in
its ability to deal with classes of shapes that are too
diverse. These classes result in priors with large parts
whose probabilities are not close to 0 or 1, and thus
the bounds obtained are not tight, reducing the effi-
cacy of the proposed method. These unspecific priors
are obtained, for example, when the parts of the shapes
in the class are not in consistent positions. To address
this problem, in Section 7.2 we divided highly vari-
able classes into subclasses with less variability. This
addressed the problem created by the highly variable
classes, at the expense of creating many more hypothe-
ses.

Another direction to extend this work that addresses
the growing number of hypotheses mentioned above, is
to exploit the redundancy among these hypotheses. It
was explained in Section 7.1 that to prove that two (or
more) hypotheses are indistinguishable, it is necessary
to refine these hypotheses completely. By doing so, how-
ever, the performance drops significantly. This problem
gets exacerbated as the number of indistinguishable hy-
potheses increases, which in turn happens when classes
are divided in subclasses or when the number of degrees
of freedom are increased. In order to exploit the redun-
dancy among hypotheses, bounds can be computed not
just for individual hypotheses but for groups of them,
_a la_ Branch & Bound. This new algorithm would then
work by splitting not just the image domain (as H&B
algorithms do), but also the space of hypotheses (as
Branch & Bound algorithms do). This new algorithm
is expected to be most effective when the hypotheses
are very similar, which is exactly the case that the cur-
rent approach is less efficient dealing with.

Acknowledgements We would like to thank Siddharth Ma-
hendran for his help running part of the experiments.
### References

1. Arrabales, R., Ledezma, A., Sanchis, A.: A cognitive approach to multimodal attention. Journal of Physical Agents 8 (2009)
2. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2006)
3. Chien, C.H., Aggarwal, J.K.: Identification of 3D objects from multiple silhouettes using quadtree/octrees. Computer Vision, Graphics, and Image Processing 36 (1986)
4. Clausen, J.: Branch and bound algorithms–principles and examples. Parallel Computing in Optimization pp. 239–267 (1997)
5. dryden.lse.co.uk: Test your awareness: Whodunnit? [http://www.youtube.com/watch?v=ubNF9QREQLA](http://www.youtube.com/watch?v=ubNF9QREQLA) (2008)
6. Dryden, I.L., Mardia, K.V.: Statistical Shape Analysis. John Wiley and Sons Ltd. (1998)
7. Elles, A.: Occupancy grids: a probabilistic framework for robot perception and navigation. PhD Dissertation, Carnegie Mellon University (1989)
8. K. F. Costa, J., Cesar, R.M.: Shape Classification and Analysis: Theory and Practice, 2nd edition edn. CRC Press (2009)
9. Frintrop, S., Rome, E., Christensen, H.I.: Computational visual attention systems and their cognitive foundations: A survey. ACM Trans. Appl. Percept. 7 (2010)
10. Fujisawa, H.: Forty years of research in character and pattern recognition. International Conference on Image Processing pp. 253–256 (2003)
11. Geman, D., Jedynak, B.: An active testing model for robot perception and navigation. PhD Dissertation, John Wiley and Sons Ltd. (1998)
12. Geman, D., Jedynak, B.: An active testing model for tracking roads in satellite images. IEEE Pattern analysis and Machine Intelligence 18, 1–14 (1996)
13. Hastie, T., Tibshirani, R., Friedman, J.: The elements of statistical learning: data mining, inference and prediction, 2nd edn. Springer (2009)
14. Kahn, D.W.: Topology: An Introduction to the Point-Set and Algebraic Areas. Dover Publications, Inc. (1995)
15. Kaheman, D.: Attention and Effort. Prentice-Hall (1973)
16. Kaplan, F., Hafner, V.: The challenges of joint attention. Interaction Studies 7, 135–169 (2006)
17. Kay, S.M.: Fundamentals of Statistical Signal Processing: Estimation Theory. Prentice Hall PTR (1993)
18. Koch, C., Ullman, S.: Shifts in selective visual attention: towards the underlying neural circuitry. Human Neurobiology 4, 219–227 (1985)
19. Koller, D.: Fundamentals of statistical learning: data mining, inference and prediction. Data Mining, 2nd edn. Springer (2009)
20. Lindeberg, T.: Scale-Space Theory in Computer Vision. Kluwer Academic Publishers (1994)
21. Lindley, D.: Making decisions. John Wiley (1985)
22. Mittal, A., Paragios, N.: Motion-based background subtraction using adaptive kernel density estimation. Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2004)
23. Naim, D., Haker, S., Bobick, A., Tannenbaum, A.: Multiscale 3D shape analysis using spherical wavelets. Proc. 8th Int. Conf. Medical Image Computing Computer-Assisted Intervention pp. 459–467 (2005)
24. Oliva, A., Torralba, A., Castellano, M.S., Henderson, J.M.: Top-down control of visual attention in object detection. International Conference on Image Processing pp. 253–256 (2003)
25. Pearl, J.: Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann Publishers, Inc. (1988)
26. Persson, E., Pu, K.S.: Shape discrimination using fourier descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence 8, 388 – 397 (1986)
27. Potmesil, M.: Generating octree models of 3D objects from their silhouettes in a sequence of images. Computer Vision, Graphics, and Image Processing 40, 1–29 (1987)
28. Rönnugren, R., Ayani, R.: A comparative study of parallel and sequential priority queue algorithms. ACM Trans. Model. Comput. Simul. 7 (1997)
29. Rosch, E.: Principles of categorization. In: E.R.B.B. Lloyd (ed.) Cognition and categorization. Erlbaum (1978)
30. Rother, D.: Supplementary data. [http://www.cis.jhu.edu/~diroth/Research/DataShapes.zip](http://www.cis.jhu.edu/~diroth/Research/DataShapes.zip) (2011)
31. Rother, D., Sapiro, G.: Seeing 3D objects in a single 2D image. Proceedings of the 12th IEEE International Conference on Computer Vision (2009)
32. Rother, D., Vidal, R.: Hypothesize and bound: A computational focus of attention mechanism for simultaneous 3D shape reconstruction, pose estimation and classification from a single 2D image. (In preparation) (2011)
33. Samet, H.: An overview of quadtrees, octrees, and related hierarchical data structures. NATO ASI Series 40, 51–68 (1988)
34. Sandhu, R., Dambreville, S., Yezzi, A., Tannenbaum, A.: Non-rigid 2D-3D pose estimation and 2D image segmentation. Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (2009)
35. Schnell, S., Mendoza, C.: A formula for integrating inverse functions. The Mathematical Gazette (2000)
36. Shic, F., Scassellati, B.: A behavioral analysis of computational models of visual attention. International Journal of Computer Vision 73, 159–177 (2007)
37. Simons, D.J., Chabris, C.F.: Gorillas in our midst: sustained inattentive blindness for dynamic events. Perception 28, 1059–1074 (1999)
38. Smitsman, R., Jedynak, B.: Active testing for face detection and localization. IEEE Pattern analysis and Machine Intelligence 32, 1914–1920 (2010)
39. Tsotsos, J., Itti, L., Rees, G.: A brief and selective history of attention. In: L. Itti, G. Rees, J. Tsotsos (eds.) Neurobiology of Attention. Elsevier Press (2005)
40. Veltkamp, R.C.: Shape matching: Similarity measures and algorithms. Shape Modelling International pp. 188–197 (2001)
41. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. Proc. IEEE Int’l Conf. Computer Vision and Pattern Recognition, Hawaii, December (2001)
42. Weiss, Y., Freeman, W.: On the optimality of solutions of the max-product belief-propagation algorithms in arbitrary graphs. IEEE Trans. on Information Theory 47, 736–744 (2001)
43. Wilcox, H.J., Myers, D.L.: An Introduction to Lebesgue Integration and Fourier Series. Dover Publications, Inc. (1978)
44. Yatziv, L., Bartesaghi, A., Sapiro, G.: A fast o(n) implementation of the fast marching algorithm. Journal of Computational Physics 212 (2006)
9 Supplementary material

Lemma 1 (Properties of LCDFs and $m$-Summaries) Let $\Pi$ be an arbitrary partition, and let $\tilde{S}$ and $D = \{D_\omega\}_{\omega \in D}$ be respectively a semidiscrete shape and a LCDF in this partition. Then:

1. Given a BF $B$ such that $B \sim D$, it holds that
   \[ \sup_{\tilde{S} \sim \tilde{S}} \int_B \delta_B(x) S(x) \, dx = \sum_{\omega \in \Pi} \int_{|\omega| - \tilde{S}(\omega)} |\omega| D_{\omega}^{-1}(1) \, da. \] \hspace{1cm} (29)

   2. Similarly, given a continuous shape $S$, such that $S \sim \tilde{S}$, it holds that
   \[ \sup_{B \sim D} \int_B \delta_B(x) S(x) \, dx = \sum_{\omega \in \Pi} \int_{|\omega| - \tilde{S}(\omega)} |\omega| D_{\omega}^{-1}(1) \, da. \] \hspace{1cm} (30)

3. Moreover, for any $\alpha \in [0, |\omega|]$, the integrals on the rhs of (29) and (30) can be bounded as
   \[ \int_{\alpha}^{|\omega|} D_{\omega}^{-1}(1) \, da \leq \frac{\delta_{\max}}{m} \left[ J(\alpha) \left( \tilde{Y}_{\omega}^{\alpha} - \alpha \right) + \sum_{j = J(\alpha)}^{m-1} \left( \tilde{Y}_{\omega}^{j+1} - \tilde{Y}_{\omega}^{j} \right) (j + 1) \right], \] \hspace{1cm} (31)

   where $J(\alpha)$ is as defined in (22).

4. The rhs of (30) and (32) are maximum when $\tilde{S}(\omega) = |\omega| - A_\omega$, thus
   \[ \sup_{\tilde{S}} \sum_{\omega \in \Pi} \int_{|\omega| - \tilde{S}(\omega)} |\omega| D_{\omega}^{-1}(1) \, da = \sum_{\omega \in \Pi} \int_{A_\omega} |\omega| D_{\omega}^{-1}(1) \, da. \] \hspace{1cm} (32)

Proof: 1. For simplicity, let us only consider the case in which the partition $\Pi$ consists of a single element $\omega$; the generalization of the proof to partitions containing multiple elements is straightforward. Let us assume for the moment that $D_\omega(\delta)$ is strictly increasing and continuous, so that $D_\omega(D_\omega^{-1}(a)) = \forall a \in [0, |\omega|]$.

   Consider the continuous shape $S^*$, defined as
   \[ S^*(x) \triangleq 1 \left( \delta_B(x) - \tilde{\delta} \right) = \begin{cases} 1, & \text{if } \delta_B(x) \geq \tilde{\delta} \\ 0, & \text{otherwise} \end{cases} \] \hspace{1cm} (55)

   where $1(\cdot)$ is the Heaviside step function and $\tilde{\delta} \triangleq D_{\omega}^{-1} \left( |\omega| - \tilde{S}(\omega) \right)$. This shape is the solution to the lhs of (29). To see this, notice that $S^* \sim \tilde{S}$ because
   \[ \int_{\omega} S^*(x) \, dx = \left\{ \{ x \in \omega : \delta_B(x) \geq \tilde{\delta} \} \right\} = |\omega| - D_\omega(\tilde{\delta}) = \tilde{S}(\omega). \] \hspace{1cm} (56)

   Notice also that $S^*$ maximizes the integral on the lhs of (29) because it contains the parts of $\omega$ with the highest values of $\delta_B$.

   Thus, substituting (55) on the lhs of (29), and defining the function $g(x) \triangleq \delta_B(x) - \tilde{\delta}$, (29) reduces to
   \[ \int \delta_B(x) 1 \left( \delta_B(x) - \tilde{\delta} \right) \, dx = \int g(x) 1 (g(x)) \, dx + \tilde{\delta} \int 1 (g(x)) \, dx. \] \hspace{1cm} (57)

   Using Lebesgue integration (79), this last expression is equivalent to
   \[ \int_{0}^{\infty} |\{ x \in \omega : g(x) \geq \tilde{\delta} \}| \, d\tilde{\delta} + \tilde{\delta} \int_{0}^{\infty} |\{ x \in \omega : g(x) \geq 0 \}| \, d\tilde{\delta} = \int_{0}^{\infty} |\{ x \in \omega : \delta_B(x) \geq \tilde{\delta} \}| \, d\tilde{\delta} + \tilde{\delta} \int_{0}^{\infty} |\{ x \in \omega : \delta_B(x) \geq \tilde{\delta} \}| \, d\tilde{\delta} = \int_{\tilde{\delta}}^{\delta_{\max}} |\omega| - D_\omega(\tilde{\delta}) \, d\tilde{\delta} + \tilde{\delta} \tilde{S}(\omega). \] \hspace{1cm} (58)

   Now recall that the integral of a function $f(x)$ can be written in terms of its inverse $f^{-1}(y)$, if it exists, as
   \[ \int_{a}^{b} f(x) \, dx = \int_{f(a)}^{f(b)} f^{-1}(y) \, dy. \] \hspace{1cm} (59)

   Hence, it follows that (58) is equal to
   \[ |\omega|[\delta_{\max} - \tilde{\delta}] - \delta_{\max} D_\omega(\delta_{\max}) + \tilde{\delta} D_\omega(\tilde{\delta}) + \int_{D_\omega(\tilde{\delta})}^{D_\omega(\delta_{\max})} D_\omega^{-1}(a) \, da + \tilde{\delta} \tilde{S}(\omega) = |\omega|[\delta_{\max} - \tilde{\delta}] - \delta_{\max} |\omega| + \tilde{\delta} (|\omega| - \tilde{S}(\omega)) + \int_{|\omega| - \tilde{S}(\omega)}^{\infty} D_\omega^{-1}(a) \, da + \tilde{\delta} \tilde{S}(\omega) = \int_{|\omega| - \tilde{S}(\omega)}^{\infty} D_\omega^{-1}(a) \, da, \] \hspace{1cm} (60)

   as we wanted to prove.

   If $D_\omega(\tilde{\delta})$ is not one-to-one (as we assumed above), the proof is essentially similar but complicated by the fact that $S^*$ can be chosen in more than one way (because there are sets $\theta \subset \omega$, with $|\theta| > 0$, where $\delta_B$ is constant) and this has to be handled explicitly.

2. To prove 2 we proceed exactly as in 1, except that in this case we choose $\delta_B$ to have its largest values in $S$, instead of choosing $S$ to be supported where the largest values of $\delta_B$ are. \(\square\)
3. From (21), it follows that
\[
\int_{a}^{\infty} D_{\omega}^{-1}(a) \, da \leq \frac{\delta_{\max}}{m} \int_{a}^{\infty} J(a) \, da = \frac{\delta_{\max}}{m} \left[ \int_{a}^{\infty} \hat{Y}_{j}^{(\infty)}(a) \, da + \sum_{j=0}^{m-1} \int_{a}^{\infty} \hat{Y}_{j}^{(m)}(j+1) \, da \right].
\]

(61)

And because the integrands on the rhs of (61) are all constant, (31) follows.

4. It is clear that each integral on the lhs of (32) is maximum when the integration domain contains only the parts where \( D_{\omega}^{-1}(a) \) is positive or zero. Therefore, from (19), each integral is maximum when computed in the interval \([A_{B,\omega}, \omega]\), which yields the rhs of (32), proving the lemma.

Theorem 1 (Lower bound for \( L(H) \)) Let \( H \) be a partition, and let \( \hat{Y}_{f} = \{ \hat{Y}_{f,\omega} \}_{\omega \in \Pi} \) and \( \hat{Y}_{H} = \{ \hat{Y}_{H,\omega} \}_{\omega \in \Pi} \) be the mean-summaries of two unknown BFs in \( \Pi \). Then, for any \( B_{f} \sim \hat{Y}_{f} \) and any \( B_{H} \sim \hat{Y}_{H} \), it holds that \( L(H) \geq L_{\Pi}(H) \), where
\[
L_{\Pi}(H) \triangleq Z_{B_{H}} + \sum_{\omega \in \Pi} L_{\omega}(H),
\]

(40)

and \( L_{\omega}(H) \triangleq (\hat{Y}_{f,\omega} + \hat{Y}_{H,\omega}) \tilde{q}_{\omega}(\omega) \),

(41)

and \( \tilde{q}_{\omega}(\omega) \) is a discrete shape in \( \Pi \) defined as
\[
\tilde{q}_{\omega}(\omega) \triangleq \begin{cases} 1, & \text{if } (\hat{Y}_{f,\omega} + \hat{Y}_{H,\omega}) > 0, \\ 0, & \text{otherwise}. \end{cases}
\]

(42)

Proof: Since the set of continuous shapes \( q \) that are compatible with the discrete shape \( \tilde{q} \) is a subset of the set of all continuous shapes (from (19)), it holds that
\[
L(H) \geq Z_{B_{H}} + \max_{q} \left[ \sup_{\tilde{q} \sim q} \int_{\Omega} (\delta_{B_{f}}(x) + \delta_{B_{H}}(x)) \, q(x) \, dx \right]
\]

(62)

\[
= Z_{B_{H}} + \max_{q} \sum_{\omega \in \Pi} \tilde{q}(\omega) \int_{\omega} (\delta_{B_{f}}(x) + \delta_{B_{H}}(x)) \, dx.
\]

(63)

The terms in the integral in (63) are, by definition, the mean-summaries of the BFs (see (23)), thus
\[
L(H) \geq Z_{B_{H}} + \max_{q} \sum_{\omega \in \Pi} \tilde{q}(\omega) \left( \hat{Y}_{f,\omega} + \hat{Y}_{H,\omega} \right).
\]

(64)

It then follows that the discrete shape defined in (42) maximizes the rhs of (61), proving the theorem.

Theorem 2 (Upper bound for \( L(H) \)) Let \( H \) be a partition, and let \( \hat{Y}_{f} = \{ Y_{f,\omega} \}_{\omega \in \Pi} \) and \( \hat{Y}_{H} = \{ Y_{H,\omega} \}_{\omega \in \Pi} \) be the m-summaries of two unknown BFs in \( \Pi \). Let \( \hat{Y}_{f} \oplus \hat{Y}_{H} (\omega \in \Pi) \) be a vector of length \( 4m + 2 \) obtained by sorting the values in \( \hat{Y}_{f,\omega} \) and \( \hat{Y}_{H,\omega} \) (in ascending order), keeping repeated values, i.e.,
\[
\hat{Y}_{f} \oplus \hat{Y}_{H} \triangleq \left[ \hat{Y}_{f,1} \oplus \hat{Y}_{H,1}, \ldots, \hat{Y}_{f,4m+2} \oplus \hat{Y}_{H,4m+2} \right].
\]

(43)

Thus, it follows that the continuous shape that maximizes (39) is equivalent to a semidiscrete shape \( \tilde{q}_{\omega} \) in \( \Pi \) that satisfies \( \tilde{q}_{\omega}(\omega) \in \{ \omega | -Y_{f,\omega} - \hat{Y}_{H,\omega} \} \forall \omega \in \Pi \).

Proof: Let \( D_{f} \) and \( D_{H} \) be two arbitrary cumulative distributions such that \( D_{f} \sim \hat{Y}_{f} \) and \( D_{H} \sim \hat{Y}_{H} \). Then,
\[
L(H) \leq Z_{B_{H}} + \sup_{B_{f} \sim D_{f}, B_{H} \sim D_{H}} \int_{\Omega} \sup_{q} \left( \delta_{B_{f}}(x) + \delta_{B_{H}}(x) \right) \, q(x) \, dx,
\]

(65)

and from (16), this expression is equal to
\[
= Z_{B_{H}} + \sup_{B_{f} \sim D_{f}, B_{H} \sim D_{H}} \int_{\Omega} \sup_{q} \left( \delta_{B_{f}}(x) + \delta_{B_{H}}(x) \right) \, q(x) \, dx.
\]

(66)

Exchanging the order of the sup operations and using (30), the rhs of (66) is less or equal than
\[
Z_{B_{H}} + \sup_{q \sim \tilde{q}} \left[ \sup_{B_{f} \sim D_{f}, B_{H} \sim D_{H}} \int_{\Omega} \delta_{B_{f}}(x) \, q(x) \, dx + \int_{\Omega} \sup_{B_{H} \sim D_{H}} \delta_{B_{H}}(x) \, q(x) \, dx \right]
\]

(67)
\[ Z_{B_H} + \sup_{\tilde{q} \in \mathcal{S}(D)} \left[ \sum_{\omega \in \Pi} \int_{|\omega| - \tilde{q}(\omega)}^{|\omega|} D_{f,\omega}^{-1}(a) \, da + \int_{|\omega| - \tilde{q}(\omega)}^{|\omega|} D_{\tilde{H},\omega}^{-1}(a) \, da \right] = \]
\[ Z_{B_H} + \sum_{\omega \in \Pi} \sup_{\tilde{q}(\omega)} \int_{|\omega| - \tilde{q}(\omega)}^{|\omega|} \left( D_{f,\omega}^{-1}(a) + D_{\tilde{H},\omega}^{-1}(a) \right) \, da. \]

(67)

Since \( D_f \sim \tilde{Y}_f \) and \( D_H \sim \tilde{Y}_H \), it follows from (21) and (22) that

\[ D_{f,\omega}^{-1}(a) + D_{\tilde{H},\omega}^{-1}(a) \leq \frac{\delta_{\max}}{m} \left( \left| \left\{ j : \tilde{Y}_f^j < a \right\} \right| + \left| \left\{ j : \tilde{Y}_H^j < a \right\} \right| - 2m \right), \]

which using (43) can be rewritten as

\[ D_{f,\omega}^{-1}(a) + D_{\tilde{H},\omega}^{-1}(a) \leq \frac{\delta_{\max}}{m} \left( \left| \left\{ j : \tilde{Y}_f^j < a \right\} \right| - 2m \right). \]

(68)

Therefore, it follows that

\[ \int_{\tilde{Y}_f \oplus H,\omega}^{\tilde{Y}_f \oplus H,\omega} (a) \, da = \]
\[ = \frac{\delta_{\max}}{m} \sum_{j = k}^{4m + 1} (j - 2m) \, da = \]
\[ = \frac{\delta_{\max}}{m} \sum_{j = k}^{4m + 1} (j - 2m) \left( \tilde{Y}_f^j \oplus H,\omega - \tilde{Y}_f^j \oplus H,\omega \right). \]

(70)

Substituting this last expression into (67) yields the bound

\[ L(H) \leq Z_{B_H} + \frac{\delta_{\max}}{m} \sum_{\omega \in \Pi} \sup_{k \in \{1, \ldots, 4m + 1\}} \sum_{j = k}^{4m + 1} (j - 2m) \left( \tilde{Y}_f^j \oplus H,\omega - \tilde{Y}_f^j \oplus H,\omega \right). \]

(71)

Since \( D_{f,\omega}^{-1}(a) \) is a non-decreasing, piecewise constant function that takes the zero value when \( a \in (\tilde{Y}_f^{2m} \oplus H,\omega, \tilde{Y}_f^{2m+1} \oplus H,\omega) \), it follows that the semidiscrete shape in (46) maximizes (67), and that the supremum in (71) is obtained when \( k = 2m + 1 \), proving the theorem. \( \square \)