Search Strategies for Multiple Landmark Detection by Submodular Maximization

David Liu, Kevin S. Zhou, Dominik Bernhardt*, Dorin Comaniciu
Siemens Corporation, Corporate Research, Princeton, NJ, USA
Siemens AG, Sector Healthcare, Computed Tomography, Forchheim, Germany*
david-liu@siemens.com

Abstract

*A fundamental issue in multiple landmark detection is the reduction of computational cost. This problem has previously been addressed mainly by reducing the complexity of each **individual** landmark detector. We address the problem by optimizing the search strategy of **multiple** landmarks. When the relative positions of landmarks are constrained, the search space can be reduced, thereby reducing the computation. The proposed method leverages the theory of submodular functions to provide a constant factor approximation guarantee of the optimal speed. Although the theory of submodular functions is well known, to the best of our knowledge, this is the first time it is applied to the landmark detection problem. We demonstrate our method by fast and accurate detection of human body landmarks including bones, organs, and vessels in 3D CT images from a diverse dataset of around 2000 volumes with pathological patients. We further provide different search space criteria and variations.*

1. Introduction

An important area in automated image understanding is the development of methods for quickly detecting a plurarity of landmarks. In image segmentation, landmarks provide seed points for the initialization of segmentation algorithms. In face recognition, landmark-based pose alignment (such as ASM and AAM) provide enhanced accuracy. In medical imaging, the registration of MRI and CT modalities uses landmarks to provide control points. In human body tracking, landmarks provide the position of individual body parts.

In this paper, we are interested in the problem of efficiently detecting a large number of anatomical landmarks in human body from CT scans. Figure 1 shows some examples of CT scans. More specifically, given a large number of detectors, how should one coordinate the detectors collaboratively and sequentially so that detection can be done most efficiently? We assume the landmark detectors have already been trained, and each detector corresponds to a distinct anatomical structure. In CT scans, each anatomical structure has at most one instance. This is different from, *e.g.*, face detection in general images, where multiple faces appear in a single image. We distinguish these two cases with the terms single-instance and multi-instance problems.

The single-instance detection problem is a constrained search problem, where the location of one landmark constrains the other ones. We present a theoretical framework based on the theory of submodular functions [20]. We show that, with properly defined search spaces, a search strategy that sequentially orders the detectors improves both speed and accuracy with theoretical guarantees.

In the sections that follow we formulate the search strategy in detail. In Section 2, we describe two search space criteria and detail the optimization procedure for efficiently detecting landmarks. In Section 3 we discuss the implementation in more detail. In Section 4 we show state-of-the-art results on a dataset of 2046 CT volumes, indicating the benefits of the method. We conclude in Section 5 with discussions on the relationship of our method with a cascade of classifiers, the use of other criteria as objectives, other related work, and future directions.

2. Proposed algorithm

The main goal of this work is to minimize the computational cost of multiple landmark detection. The computational cost is controlled by (i) the size of the image subspace (or *search space*) in which a detector is performing the search, and (ii) the unit cost of the landmark detector. We will first focus on item (i), and later extend the framework to item (ii).

Having n landmarks detected, with $N-n$ landmarks remaining to detect, which detector should one use next, and where should it be applied, so that the overall computational cost is minimized? These two questions are tightly related,

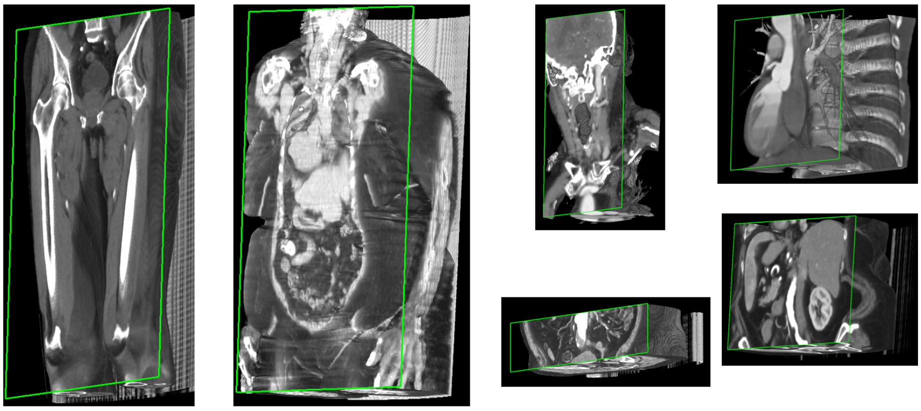

Figure 1. Our dataset consists of 3D CT scans from different body portions, including larger scans shown in column 1 and 2, head scans and lower abdomen scans in column 3, heart and abdomen scans in column 4. Automatic detection of body parts including organs, bones, and vessels facilitates applications such as automatic segmentation and clinical measurements.

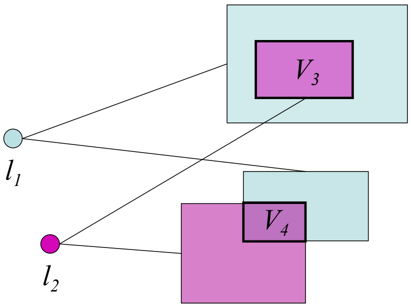

Figure 2. Illustration of the search space definition in Eq.(2). Detected landmarks l_1 and l_2 provide search spaces for un-detected landmarks l_3 and l_4 (not shown). Final search spaces V_3 and V_4 for l_3 and l_4 are obtained by intersection. A greedy algorithm would prefer landmark l_4 over l_3 as the next landmark to detect since V_4 is smaller than V_3.

and our answer is simple: *determine the search space for each detector based on the already detected landmarks and pick the detector that has the smallest search space.* We will show theoretical guarantees of the algorithm in Section 2.2, and then in Section 2.3 extend the algorithm to take multiple factors into account, including the size of the search space and the unit cost of the detector (classifier).

2.1. Search space

In sequential detection, landmarks already detected provide spatial constraints on the landmarks remaining to be detected. Consider an object consisted of N distinct landmarks. Denote by

$$\Lambda_{(1):(n)} = \{l_{(1)} \prec l_{(2)} \prec ... \prec l_{(n)}\}, n \leq N, \tag{1}$$

the ordered set of detected landmarks. Denote by U the unordered set of landmarks that remains to be detected. For each landmark $l_i \in U$, its search space Ω_{l_i} is determined jointly by landmarks in $\Lambda_{(1):(n)}$, *e.g.*, by the intersection of the individual search spaces,

$$\Omega_{l_i}(\Lambda_{(1):(n)}) = \bigcap_{j, l_j \in \Lambda_{(1):(n)}} \Omega_{l_i}(\{l_j\}), \tag{2}$$

where $\Omega_{l_i}(\{l_j\})$ denotes the search space for landmark l_i conditioned on the position of a detected landmark l_j. This is illustrated in Figure 2. This definition could be restrictive, so we will discuss alternatives in Section 2.3 and 2.4.

Denote the search volume (or search area) of search space $\Omega_{l_i}(\Lambda)$ as $V(\Omega_{l_i}(\Lambda))$, which calculates the volume of $\Omega_{l_i}(\Lambda)$. Without loss of generality, assume the search volume is the cardinality of the set of voxels (pixels) that fall within the search space. Define the constant $\Omega_\phi \equiv \Omega_k(\phi), \forall k$, as the space of the whole image, which is a tight upper bound of the search space. The search volume has the following property:

Theorem 1. $\forall S \subseteq T$,

$$V(\Omega(S)) - V(\Omega(S \cup \{l\})) \geq V(\Omega(T)) - V(\Omega(T \cup \{l\})) \tag{3}$$

We provide the proof in the Appendix. Set functions satisfying the above property are called *supermodular* [20].

Our goal is then to find the ordered set $\Lambda_{(2):(N)}$ that minimizes the cumulated search volume, i.e. ,

$$\Lambda'_{(2):(N)} = \underset{\Lambda_{(2):(N)}}{\operatorname{argmin}} \sum_{i=2}^{N} V(\Omega_{l_{(i)}}(\Lambda_{(1):(i-1)})). \tag{4}$$

Note that in Eq.(4) we do not include the first landmark l_1 as its search space is typically the whole image when no landmarks have been detected a priori. We will discuss the issue of determining $l_{(1)}$ later in Section 3.1.

2.2. Greedy algorithm

Define the cost function $C_k(\Lambda) = V(\Omega_k(\Lambda)), \forall k$. A greedy algorithm for finding the ordering $\{l_{(1)}, ..., l_{(N)}\}$ that attempts to minimize the overall cost proceeds as follows:

```
Initialize $\Lambda = \{l_{(1)}\}$
for j=2,...,N do
    $l_{(j)} = \arg\min_k C_k(\Lambda_{(1):(j-1)})$
    Append $l_{(j)}$ to the ordered set $\Lambda_{(1):(j-1)}$
end
```

In other words, in each round one selects the detector that yields the smallest cost.

This simple algorithm has nice theoretical properties. Define

$$F_k(\Lambda) = C_k(\phi) - C_k(\Lambda) \tag{5}$$

Hence, $F_k(\phi) = 0$. In the Appendix (Lemma 7) we show that $F_k(.)$ is a nondecreasing set function. From Eq.(1) and (5), $\forall S \subseteq T$,

$$F_k(S) - F_k(S \cup \{l\}) \leq F_k(T) - F_k(T \cup \{l\}) \tag{6}$$

which means $F_k(.)$ is submodular [20]. Furthermore, since $C_k(\phi)$ is constant over k, Eq.(4) becomes

$$\Lambda'_{(2):(N)} = \underset{\Lambda_{(2):(N)}}{\operatorname{argmax}} \sum_{k=2}^{N} F_k(\Lambda_{(1):(k-1)}). \tag{7}$$

Lemma 1. *$F(.) = \sum F_k(.)$ is submodular if $\forall k, F_k(.)$ is submodular [20].*

Together, these properties bring us to the theorem that states the theoretical guarantee of the greedy algorithm.

Theorem 2. *If $F(.)$ is a submodular, nondecreasing set function and $F(\phi) = 0$, then the greedy algorithm finds a set Λ', such that $F(\Lambda') \geq (1 - 1/e) \max F(\Lambda)$ [17].*

Optimizing submodular functions is in general NP-hard [16]. One must in principle calculate the values of $N!$ detector ordering patterns. Yet, the greedy algorithm is guaranteed to find an ordered set Λ such that $F(.)$ reaches at least 63% of the optimal value.

Note that the ordering found by the algorithm is image-dependent, since the search space of the next detector is dependent on the position of the landmarks already detected. Therefore, the algorithm is not performing an 'off-line' scheduling of detectors. For another example, when the search space of a landmark is outside the image or if its detection score is too low, then this landmark is claimed missing. This would influence the subsequent detectors through the definition of the search space and affect the final ordering.

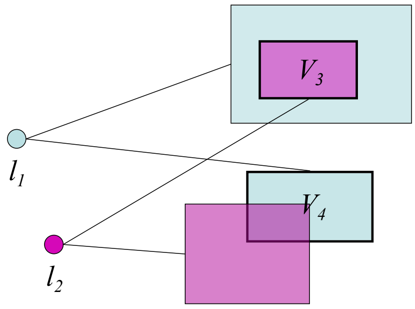

Figure 3. Illustration of the search space definition in Eq.(8). Detected landmarks l_1 and l_2 provide search spaces for un-detected landmarks l_3 and l_4 (not shown). Final search spaces V_3 and V_4 for l_3 and l_4 are the minimum sets. This time, a greedy algorithm would prefer landmark l_3 over l_4 as the next landmark to detect since V_3 is smaller than V_4.

2.3. Another search space criteria

Another useful definition of search space can be defined as follows:

$$\Omega_{l_i}(\Lambda) = \min_{l \in \Lambda}\{\Omega_{l_i}(l)\} \tag{8}$$

In each round of the greedy algorithm, each detected landmark provides a search space candidate for each un-detected landmark. Each un-detected landmark then selects the smallest one among the provided candidates. The greedy algorithm then selects the un-detected landmark that has the smallest search space. This is illustrated in Figure 3. We call this search space criteria the *min-rule*, and the one in Section 2.1 the *intersection-rule*. In the Appendix we show that the min-rule also satisfies supermodularity.

2.4. Multiple search space criteria

Since submodularity is closed under linear combination with non-negative scalars [20], multiple submodular functions can be optimized simultaneously. For example, one could combine the min-rule and intersection-rule. Note that the set of individual search spaces $\{\Omega_{l_i}(\{l_j\})\}_{i,j=1,...,N}$ need not be the same for the min- and intersection-rules. Under this combination, some detectors could obtain a search range from the min-rule, and some from the intersection-rule. [1] [2]

2.5. Cost of Detector

The algorithm introduced so far only considered the search space. In practice, different detectors have different costs and this should be taken into account during opti-

[1] If $\Omega_{l_i}(\{l_j\}), \forall i, j$, were the same for min- and intersection-rules, then the intersection-rule will always be selected, since the intersection operation yields non-larger spaces than individual ones.

[2] Linear combination is a common approach for finding Pareto-optimal solutions [2]. Since it can happen that $F_i(S) > F_i(T)$ while $F_j(T) > F_j(S)$, all we can hope for are Pareto-optimal solutions [2].

mization. For example, if we have two detectors, then the algorithm above would select the next detector that has a smaller search space. However, this detector might have a much higher unit computational cost due to, *e.g.*, higher model complexity. One should multiply (and not linearly combine) search volume with the unit cost, since a detector is applied to *each* voxel within the search space and only the product reflects the cost correctly.

Fortunately, multiplication of a submodular function by a non-negative scalar also maintains submodularity [20]. Denote q_i as the computational cost of detector i. The product $q_i C(\Omega_{l_i}(\Lambda))$ then considers the joint computational cost. Since $\forall i, q_i \geq 0, q_i C(\Omega_{l_i}(\Lambda))$ is submodular, the greedy algorithm can be applied and the same theoretical guarantees still hold.

The computational cost of a classifier can be estimated from, *e.g.*, the number of weak learners in boosting-based classifiers, the expected number of classifiers in a cascade of classifiers [22], or the empirical running time. We only need the cost up to a constant multiplicative factor k since q_i and kq_i yield the same results.

3. Implementation

3.1. Occlusion and the anchor landmark

The algorithm in the previous section finds an image-dependent ordering of detectors assuming (any) one landmark $l_{(1)}$ has already been detected. We call $l_{(1)}$ the *anchor landmark*. Note that the anchor landmark can be a different landmark (body part) in different images. Finding the anchor landmark might require multiple trials when the earlier detection trial(s) claims false negative or when some landmarks are occluded or missing (such as in CT partial scans, see Figure 1). Multiple trials require multiple searches over the whole image and are expensive.

This suggests a priority queue of landmarks sorted by conditional frequency. Define $f(l)$ as the estimated frequency of appearance of landmark l in an image. Then, define the ordering of trials

$$m_1 = \arg\max_l \{f(l_1), ..., f(l_N)\} \tag{9a}$$

$$m_2 = \arg\max_l \{f(l_1), ..., f(l_N) | m_1 \text{ not present}\} \tag{9b}$$

$$m_3 = \arg\max_l \{f(l_1), ..., f(l_N) | m_1, m_2 \text{ not present}\} \tag{9c}$$

and so on. We use this ordering of trials to detect the anchor landmark. Intuitively, since landmark m_1 appears most frequently, searching for it in the first trial would reduce most significantly the need for a subsequent trial (whole-image search). Landmark m_2 is the most frequent landmark under the condition that m_1 does not exist in the volume. This conditioning is to avoid m_2 being a landmark that is in the vicinity of m_1, in which case if m_1 is occluded, most likely m_2 is also occluded.

Since all of our detectors have similar accuracy and computational cost, such an ordering based on conditional frequency performs well. However, if some detectors have very different accuracy or cost than the others, those characteristics should also be taken into account.

The system starts with detecting the anchor landmark and initiates the greedy algorithm. If the greedy algorithm determines a search space but the corresponding detector fails to find the landmark, the greedy algorithm simply proceeds to the next round. If all subsequent landmarks are not found, the system is restarted with a different anchor landmark. The chance that the system produces more false positives than running the detectors independently is low. This is because, while the false positive rate of each detector could be high, the chance that multiple detectors produce false positives within their assigned search spaces is exponentially low. In fact there is a relationship between the overall false positive rate, detection rate, and the size of the individual search spaces. We have experiments and discussions on this topic in Section 5.1.

3.2. Coarse-to-fine

In earlier discussions, we assumed each landmark is associated with a single detector. In our implementation, a landmark has $R = 3$ detectors, each trained at different resolutions. Since training of landmark detectors is not the focus of this paper, we omit the details. In detection, we employ a coarse-to-fine strategy. Such multi-resolution techniques are frequently encountered when the solution to the original (high) resolution is either too complex to consider directly or is subject to large numbers of local minima. The general idea is to construct approximate, coarser versions of the problem and to use the solution of the coarser problems to guide solutions at finer scales.

We run the algorithm in Section 2 using the coarsest-resolution detectors only. We then define a local (small) search space around each detected landmark and run higher resolution detectors within the local search space. The overall approach is efficient, because the coarse-resolution detectors have already rejected most of the voxels in the image.

At the end, the posterior probability of position x is taken from all resolutions using a log-linear model

$$p(x|I_{r_1}, ..., I_{r_R}) \propto \exp(\sum\nolimits_{i=1}^{R} \alpha_{r_i} \phi_{r_i}(x)) \tag{10}$$

where I_{r_i} is the volume at resolution r_i, $p(x|I_{r_i})$ is the posterior probability from the detector with resolution r_i, and the potential functions are given by $\phi_{r_i}(x) = \log p(x|I_{r_i})$. This can be shown equivalent to a products-of-experts model [11]. We also experimented with the mixture-of-experts model [12] of the form

$$p(x|I_{r_1}, ..., I_{r_R}) \propto \sum\nolimits_{i=1}^{R} \alpha_{r_i} p(x|I_{r_i}). \tag{11}$$

	mean	std	Q95	max
Independent D_{8mm} $N = 63$	17.30	6.16	46.24	84.51
Greedy D_{8mm} $N = 63$	1.14	0.47	1.92	2.44
Independent D_{8mm} $N = 25$	6.72	6.40	17.73	35.00
Greedy D_{8mm} $N = 25$	0.65	0.43	1.26	5.08
Greedy D_{4mm} $N = 25$	1.30	0.87	3.30	6.11
Greedy D_{2mm} $N = 25$	2.70	1.74	7.15	9.05

Table 1. Detection time (sec) per volume. N is the number of landmarks in the system.

	D_{8mm}	D_{4mm}	D_{2mm}
TracheaBif.	11.9	3.6	2.8
L.LungTop	22.2	3.5	3.2
R.LungTop	13.8	3.7	3.7
LiverDome	14.8	3.4	2.9
L.Kidney	13.6	6.7	6.3
R.Kidney	15.1	5.6	7.0

Table 2. Mean distance error in millimeters.

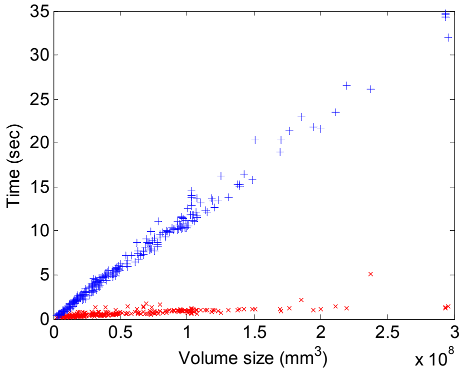

Figure 4. Detection time as a function of volume size. Blue (+): independent landmark detectors. Red (x): Greedy search.

While the products-of-experts tends to produce sharper classification boundaries, the mixture-of-experts tends to have a higher tolerance to poor probability estimates [13]. Our experiments suggest the use of the mixture-of-experts.

Notice that, each position is associated with multiple subwindows at different resolutions, so global context is utilized. Using global context is shown in many work. Our approach combines results from multiple resolutions and different subwindow sizes and hence is different from approaches where a single, optimal window size is determined [23].

In summary, we use coarse resolution detectors in a sequential ordering (which can be considered as a 'spatial' cascade of classifiers) to quickly identify position estimates for each landmark. Higher-resolution detectors subsequently refine the position estimates within a local neighborhood.

4. Experiments

Our dataset consists of 2046 volumes. We split the data into 70% training, 10% validation, and 20% testing, while avoiding splitting a patient with multiple scans into both training and testing. The pairwise search spaces, $\Omega_{l_i}(\{l_j\})$, for each pair of landmarks l_i, l_j, are cuboids estimated from training and validation data. Using only training data to define a tight cuboid for one landmark given another could result in too confined search spaces if the detectors have large errors in testing. We obtain this error information from the validation set and enlarge the cuboids accordingly.

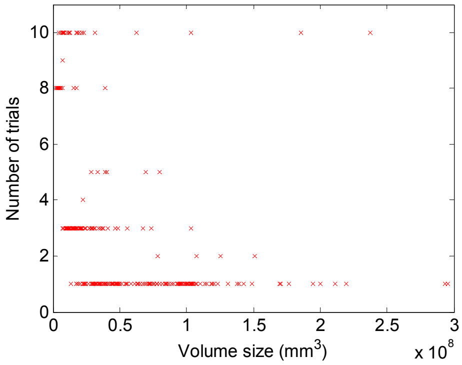

Figure 5. Number of trials to find the anchor landmark as a function of volume size.

We have 63 landmarks including positions such as the center, top, and bottom of organs, bones, and bifurcations of vessels. In Table 1 we show the speed of landmark detection when all landmarks are detected independently versus the proposed method with the min-rule. Q95 is the 95^{th} percentile. D_{8mm} denotes the system using detectors trained at 8mm resolution running the greedy method (without coarse-to-fine), D_{4mm} uses the coarse-to-fine strategy with 8 and 4mm resolution detectors, and D_{2mm} uses detectors at all three (8,4, and 2mm) resolutions.

Table 1 also includes experiments where only a subset of 25 landmarks are used in the system. We observe that the detection time of the greedy approach is not linearly proportional to the number of landmarks. In fact, when using 'fewer' landmarks, the maximum time 'increased' from 2.44 to 5.08 sec. This can be understood because the search space of each detector is provided by the landmarks already detected, some of which are not present in the 25-detector system.

The detection speed versus volume size is shown in Figure 4. The reason that smaller volumes do not consume much less time can be understood from Figure 5, which shows that smaller volumes often require more number of trials to find the anchor landmark. Since each trial requires a whole image search, detection time increases.

The detection errors of the coarse-to-fine strategy are

		Q25	Q50	Q95
Iliac Bifurcation	Baseline	6.76	13.96	22.72
	Product	4.67	8.43	**11.56**
	Mixture	**3.95**	**7.06**	13.99
Brachioc. Artery	Baseline	4.20	5.46	9.89
	Product	4.30	6.83	11.46
	Mixture	**3.63**	**5.19**	**8.77**

Table 3. Distance errors in millimeters comparing coarse-to-fine detection using the product- and mixture-of-experts.

shown in Table 2.

Two comparisons to recent literature can be made. First, the work in [4] reported a detection time around 2 sec for 9 landmarks, and mean distance error around 28 mm. Our system achieves lower distance errors in less time even on a standard Intel Core2 Duo CPU 2.66GHz (whereas they used a GPU implementation for speed up). Second, the work in [24] reported larger distance errors (kidney error 9mm, versus our 6mm using coarse-to-fine 8mm- and 4mm-resolution detectors) with detection time around 4 sec for 6 landmarks, significantly slower than our system (1.3 sec for 25 landmarks using coarse-to-fine 8mm- and 4mm-resolution detectors).

In Table 3 we compare the different coarse-to-fine approaches discussed in Section 3.2. The baseline approach finds the top candidates at one resolution and initiates a finer-resolution detection around those top candidates. This has a shifting problem (much like in visual tracking) when only neighboring resolutions are considered and information from the earliest resolutions are lost.The mixture-of-experts often has the most accurate results and has reasonable tolerance to outliers.

Some detection results of vessels are shown in Figure 6. Diseased vessels have high appearance variations, and yet we detect the carotid, iliac, renal, and brachiocephalic bifurcations with mean error 3.9, 7.2, 5.2, and 5.3 mm, respectively.

Table 4 shows confusion matrices of 8 mm detectors. This will be discussed further in Section 5.1.

5. Discussion and related work

5.1. A spatial cascade of classifiers

One might worry that a sequential detection approach could break down if the anchor landmark is incorrect or the first few detectors fail. Furthermore, the proposed search strategy was driven by computational cost considerations, and accuracy in terms of false positive rate and detection rate was not mentioned. Here we argue that the sequential 'accept or reject' behavior of our method behaves similar to a Viola-Jones cascade of classifiers [22]. Intuitively, while the false positive rate of the first detector could be high, the rate that the first n detectors all fail is significantly lower.

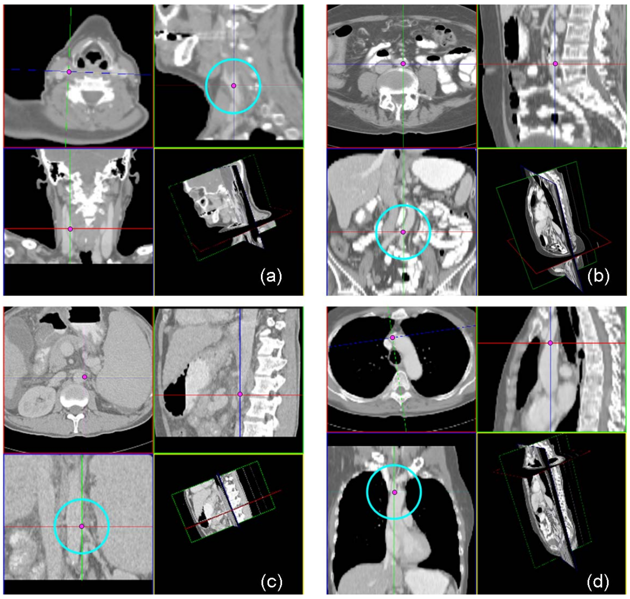

Figure 6. Detected results of (a) carotid (b) iliac (c) renal (d) brachiocephalic artery bifurcations.

	FP_A	FN_A	FP_B	FN_B
SkullBase	**0 (193)**	0 [50]	1 (192)	0 [50]
R.LungTop	**0 (84)**	1 [114]	1 (83)	1 [114]
LiverDome	0 (86)	2 [65]	0 (86)	2 [65]
R.HipTip	**0 (131)**	0 [94]	1 (130)	0 [94]
R.Knee	0 (265)	0 [12]	0 (265)	0 [12]
LiverBott.	2 (33)	1 [33]	2 (33)	1 [33]
TracheaBif.	0 (44)	0 [41]	0 (44)	0 [41]
LiverCent.	**0 (90)**	1 [136]	2 (88)	1 [136]
L.HumerusHead	0 (96)	1 [12]	0 (96)	1 [12]
R.HumerusHead	1 (80)	2 [7]	1 (80)	2 [7]
L.LungTop	**0** (61)	1 [21]	1 (61)	1 [20]
L.HipTip	**0 (94)**	**1 [46]**	2 (92)	2 [45]
L.FemurHead	0 (124)	0 [16]	0 (124)	0 [16]
R.FemurHead	0 (120)	0 [16]	0 (120)	0 [16]
CoccyxTip	0 (118)	0 [16]	0 (118)	0 [16]
PubicSymph.Top	0 (133)	0 [23]	0 (133)	0 [23]
SternumTip	3 (51)	1 [22]	3 (51)	1 [22]
AortaBend	**0 (31)**	1 [53]	1 (30)	1 [53]
Brachioceph.	1 (35)	3 [132]	1 (35)	3 [132]
R.Kidney	2 (59)	5 [61]	2 (59)	5 [61]
L.Kidney	0 (71)	0 [76]	0 (71)	0 [76]

Table 4. Confusion matrices. The first two columns (with subscript A) show the number of false positives and false negatives of our approach. Numbers in parentheses are the number of true negatives. Numbers in brackets are the number of true positives. The last two columns uses detectors running independently. Detections with distance error larger than 5 voxels are false positives.

More formally, if each detector has false positive rate f_i and detection rate d_i, the overall false positive rate and detection rate are $f = \prod f_i$ and $d = \prod d_i$ assuming independence. But f and d depend on the size of the search space,

$\Omega_{l_i}(.)$. With a slight abuse of annotation, assume search space Ω_{l_i} is a cuboid, and $\lambda\Omega_{l_i}(.), \lambda > 0$, is an enlarged or shrunk cuboid with the same center. The operating point of the ROC curve can then be adjusted by tuning λ. As λ increases, the individual detectors behave more independently and there is less cascade-effect. As λ decreases, d_i and f_i decrease, and so do d and f. Tuning individual classifiers to adjust the overall f and d is also presented in the Viola-Jones cascade of classifiers.

On the other hand, in the Viola-Jones cascade, classifiers of the 'same' landmark are chained together. Here, classifiers of 'different' landmarks are chained and provide robustness through their joint spatial relationship. As shown in Table 4, this geometric cascade indeed reduces false positives without sacrificing the detection rate. Such a robustness property is desirable and is typically implemented by random fields [1] or voting procedures [5]. If desired, one can still enforce a random field or perform voting on top of our method.

5.2. Submodularity

The problem of maximizing a submodular function is of central importance, with special cases including Max Cut [9], maximum facility location [3]. While the graph Min Cut problem is a classical polynomial-time solvable problem, and more generally it has been shown that any submodular function can be 'minimized' in polynomial time, maximization turns out to be an NP-hard problem [19]. This paper is the first one to apply the theory of submodularity to object detection. More specifically, we prove the submoduarity of the cost functions based on two search space criteria.

5.3. Information gain and entropy

Several works use maximization of information gain or entropy as objective function [14][18][24]. However, using information gain or entropy as objective could actually lead to arbitrary bad computation time!

Assume we have three landmarks, A, B, and C, with position x_A, x_B, x_C distributed along a 1-D line with position parameters $(\mu_A = 0, \Sigma_{AA} = 1), (\mu_B = 10, \Sigma_{BB} = 30), (\mu_C = 10, \Sigma_{CC} = 110)$, and $\Sigma_{AB} = 5, \Sigma_{AC} = 110, \Sigma_{BC} = 50$. This distribution could model the height of different people, with landmark A aligned with the CT scanner. Assume landmark A has already been detected. Which landmark should one detect next? The approach in [24] selects C since it yields a higher information gain than B. However, if the size of the search spaces of B and C are positively correlated with conditional covariance $\Sigma_{B|A}$ and $\Sigma_{C|A}$, the search space of B is actually smaller than the search space of C. Without considering other factors, this means the decision based on search space will be contrary to the one based on information gain. With different covariance matrices, the difference could be arbitrarily large. This can be understood when we realize that the objective of maximizing information gain does not have a direct relationship with saving computation time.

The advantages of information gain mentioned in those work, however, should not be neglected. Therefore finding a framework for gracefully trading-off between information gain and computation time would be useful.

5.4. Speed ups

Other methods for reducing computational cost in object detection include tree-structured search [10], coarse-to-fine detection [8], cascade of classifiers [22], branch-and-bound [15], reduction of classifier measurements [21], and searching in marginal space [25].

5.5. Multiple instances

In medical imaging, most anatomical structures have distinct appearances and separately trained detectors and hence detection is a single-instance problem. Real world image datasets such as the PASCAL dataset often contain multiple instances of the same class (although datasets such as Caltech-101 are single-instance). For multiple instances, our algorithm can be embedded in a parts-based framework such as [7][6] to speed up the search for object-parts.

5.6. Scaling up

Our goal is to detect in the order of thousands of anatomical structures. With such a large number of detectors, the computational savings of our approach would be significant.

References

[1] J. Besag. On the statistical analysis of dirty pictures. *Journal of Royal Statistical Soc.*, B-48:259–302, 1986.

[2] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge Univ. Press, 2004.

[3] G. Cornuejols, M. Fischer, and G. Nemhauser. On the uncapacitated location problem. *Annals of Discrete Math*, 1:163–178, 1977.

[4] A. Criminisi, J. Shotton, and S. Bucciarelli. Decision forests with long-range spatial context for organ localization in CT volumes. In *MICCAI workshop on Probabilistic Models for Medical Image Analysis*, 2009.

[5] R. O. Duda and P. E. Hart. Use of the Hough transformation to detect lines and curves in pictures. *Comm. ACM*, 15:11–15, 1972.

[6] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Trans. Pattern Analysis and Machine Intelligence*, preprint.

[7] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *Intl. Journal of Computer Vision*, 61(1):55–79, 2005.

[8] F. Fleuret and D. Geman. Coarse-to-fine face detection. *Intl. Journal of Computer Vision*, 41:85–107, 2001.

[9] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42:1115–1145, 1995.

[10] W. Grimson. *Object Recognition by Computer: The Role of Geometric Constraints*. MIT Press, 1990.

[11] G. Hinton. Products of experts. In *Intl. Conf. Artificial Neural Networks (ICANN)*, 1999.

[12] R. Jacobs, M. I. Jordan, N. S. J., and G. E. Hinton. Mixtures of expert networks. *Neural Computation*, 3:79–87, 1991.

[13] J. Kittler, M. Hatef, R. Duin, and J. Matas. On combining classifiers. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20:226–239, 1998.

[14] A. Krause and C. Guestrin. Near-optimal nonmyopic value of information in graphical models. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2005.

[15] C. H. Lampert, M. B. Blaschko, and T. Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *IEEE Conf. Computer Vision and Pattern Recognition*, 2008.

[16] L. Lovasz. *Submodular Functions and Convexity*, pages 235–257. Springer, 1983.

[17] G. Nemhauser, L. Wolsey, and M. Fisher. An analysis of the approximations for maximizing submodular set functions. *Mathematical Programming*, 14:265–294, 1978.

[18] N. Roy and C. Earnest. Dynamic action spaces for information gain maximization in search and exploration. In *American Control Conference*, pages 6–11, 2006.

[19] A. Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theory*, B(80):346–355, 2000.

[20] A. Schrijver. *Combinatorial Optimization, Polyhedra and Efficiency*. Springer, 2003.

[21] J. Sochman and J. Matas. Waldboost - learning for time constrained sequential detection. In *IEEE Conf. Computer Vision and Pattern Recognition*, 2005.

[22] P. Viola and M. Jones. Robust real-time face detection. *Intl. Journal of Computer Vision*, 57:137–154, 2004.

[23] P. Viola, J. Platt, and C. Zhang. Multiple instance boosting for object detection. *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2005.

[24] Y. Zhan, X. Zhou, Z. Peng, and A. Krishnan. Active scheduling of organ detection and segmentation in whole-body medical images. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2008.

[25] Y. Zheng, B. Georgescu, H. Ling, S. Zhou, M. Scheuering, and D. Comaniciu. Constrained marginal space learning for efficient 3D anatomical structure detection in medical images. In *IEEE Conf. Computer Vision and Pattern Recognition*, 2009.

6. Appendix

6.1. Intersection-rule

In Section 2.1 we defined $\Omega_{l_i}(S) = \bigcap_{l \in S} \Omega_{l_i}(\{l\})$. Let us simplify the notation by omitting the subscript l_i from Ω_{l_i}. Define the complement $\overline{\Omega(S)} = \Omega_\phi \setminus \Omega(S), \forall S$, where Ω_ϕ has earlier been defined as the space of the whole volume.

Lemma 2. $\Omega(S \cup \{l\}) = \Omega(S) \cap \Omega(\{l\})$

This follows from the definition.

Lemma 3. *If* $S \subseteq T$*, then* $\Omega(S) \supseteq \Omega(T)$

Proof: $T = S \cup (T \setminus S) \Rightarrow$ From Lemma 2, $\Omega(T) = \Omega(S \cup (T \setminus S)) = \Omega(S) \cap \Omega(T \setminus S) \subseteq \Omega(S)$.

Lemma 4. $\Omega(S) \setminus \Omega(S \cup \{l\}) = \Omega(S) \cap \overline{\Omega(\{l\})}$

Proof: LHS = $\Omega(S) \setminus (\Omega(S) \cap \Omega(\{l\})) = \Omega(S) \cap \overline{(\Omega(S) \cap \Omega(\{l\}))} = \Omega(S) \cap (\overline{\Omega(S)} \cup \overline{\Omega(\{l\})}) = (\Omega(S) \cap \overline{\Omega(S)}) \cup (\Omega(S) \cap \overline{\Omega(\{l\})})$ = RHS.

Lemma 5. *If* $\Omega(T) \subseteq \Omega(S)$*, then* $V(\Omega(S) \setminus \Omega(T)) = V(\Omega(S)) - V(\Omega(T))$

Lemma 6. *If* $\Omega(T) \subseteq \Omega(S)$*, then* $V(\Omega(T) \leq V(\Omega(S))$

Finally we prove the supermodularity of $V(\Omega(.))$ in Theorem 1.

Theorem 1. $\forall S \subseteq T$,

$$V(\Omega(S)) - V(\Omega(S \cup \{l\})) \geq V(\Omega(T)) - V(\Omega(T \cup \{l\}))$$

Proof of Theorem 1: From Lemma 3, $\Omega(S) \supseteq \Omega(T)$. Then $\Omega(S) \cap \overline{\Omega(\{l\})} \supseteq \Omega(T) \cap \overline{\Omega(\{l\})}$. From Lemma 4, we have $\Omega(S) \setminus \Omega(S \cup \{l\}) \supseteq \Omega(T) \setminus \Omega(T \cup \{l\})$. From Lemma 6, we have $V(\Omega(S) \setminus \Omega(S \cup \{l\})) \geq V(\Omega(T) \setminus \Omega(T \cup \{l\}))$. From Lemma 5, Q.E.D.

Lemma 7. *F(.) in Eq.(5) is nondecreasing.*

Proof: From Lemma 3 and Lemma 6, we have $\forall S \subseteq T$, we have $V(\Omega(T)) \leq V(\Omega(S))$, which shows $V(.)$ is nonincreasing. Consequently, $F(.)$ is nondecreasing.

6.2. Min-rule

In Section 2.3 we defined $\Omega_{l_i}(S) = \min_{l \in S}\{\Omega_{l_i}(l)\}$. Denote $m_{l_i}(S) = \mathrm{argmin}_{l \in S}\, \Omega_{l_i}(\{l\})$ as the landmark in the set of detected landmarks S that provides the smallest search range for detector l_i. Here we show that this definition also satisfies Theorem 1.

Lemma 8. $\forall S \subseteq T, V(\Omega(S)) \geq V(\Omega(T))$

Proof: From definition, $\Omega(S) = \min_{l \in S} \Omega(\{l\})$, and $\Omega(T) = \min_{l \in T} \Omega(\{l\})$. Since $S \subseteq T$, we have $\Omega(S) \geq \Omega(T)$, and hence $V(\Omega(S)) \geq V(\Omega(T))$.

Proof of Theorem 1:
Case (i): $m_{l_i}(T) = m_{l_i}(T \cup \{l\})$. This means including l does not decrease the search space, and hence $V(\Omega(T)) = V(\Omega(T \cup \{l\}))$. But from Lemma 8, $V(\Omega(S)) \geq V(\Omega(S \cup \{l\}))$ always holds. Hence $V(\Omega(S)) - V(\Omega(S \cup \{l\})) \geq V(\Omega(T)) - V(\Omega(T \cup \{l\}))$.

Case (ii): $m_{l_i}(T) \neq m_{l_i}(T \cup \{l\})$. This means l provides a smaller search space than any other landmark in T, and hence $m_{l_i}(\{l\}) = m_{l_i}(T \cup \{l\})$. Since $S \subseteq T$, we also have $m_{l_i}(\{l\}) = m_{l_i}(S \cup \{l\})$. Hence, $V(\Omega(S \cup \{l\})) = V(\Omega(T \cup \{l\}))$. But from Lemma 8, $V(\Omega(S)) \geq V(\Omega(T))$ always holds. Hence $V(\Omega(S)) - V(\Omega(S \cup \{l\})) \geq V(\Omega(T)) - V(\Omega(T \cup \{l\}))$.