Strategies for Active Sensor Management in Active Sensors for Local Planning in Mobile Robotics (Ed. P. J. Probert Smith)

Steven Reece
Robotics Research Group,
Dept. Engineering Science
Oxford University
Parks Road
Oxford
OX1 3PJ, UK
reece@robots.ox.ac.uk

Abstract
Our ability to devise safe, reliable multi-sensor systems is critical in an age where sensor platforms are increasingly complex. The difficulties inherent in the design and verification of complex systems necessitates intelligent reconfiguration of sensors and their signal processing tools online. This paper introduces a self-adaptive (i.e. deliberative) approach to flexible model-based image understanding. The methods described in this paper use qualitative reasoning to maintain probabilistic models of qualitative sensor-centred scene descriptors.

1 Introduction
This chapter focuses on the need to adapt and schedule signal processing software and sensors for image interpretation tasks. A framework is introduced by which a set of sensors and/or algorithms can adapt to different environments and contexts. The chapter addresses the structure applied to signal processing software.

Our ability to devise safe, reliable signal processing software is critical in an age where sensor platforms are increasingly sophisticated. System robustness is hard to guarantee as it is virtually impossible to anticipate all the rich,
Figure 1: An Oxford park setting (including bench, tree and path) seen through a 4.0 to 6.0 micron infrared camera on (1, 2 and 3) a bright day, (4) a dull day, (5) a rainy day, and (6) at night on January 27, 2000. Images were obtained with an Inframetrics, InfraCAM infrared camera, courtesy British Aerospace.
subtle and complex operating environments the system may encounter. For the sensor data to be interpreted correctly, significant experience or knowledge of the physics of the sensing process must be available. For example, Fig. 1 shows how lighting and temperature conditions can affect infrared images. The human system designer, who may be familiar with the intricacies of vision, may not be completely familiar with the behaviour of sensor modalities such as infrared and ultrasound. Such modalities are best described using very unfamiliar, sensor-centred concepts. For example, a recent ontology to emerge for describing specular images obtained by scanning, time-of-flight ultrasound sensors contains the “Region of Constant Depth” (RCD) concept [11], which is a feature common to all sonar images involving specular reflectors but does not occur in vision. There are many sensor-centred concepts, such as the thermal shadow in infrared sensing.

In general, a sensor ontology contains fundamental and compound concepts. A fundamental concept is a basic sensor-centred feature found in many environment images (the RCD described above, for example). Compound concepts are invariant within an environment but may change between environments. For example, cities, parks and airports are compound concepts of urban environments whereas forests, lakes and mountains are compound concepts of natural environments. Compound concepts are described by combinations of fundamental concepts and their behaviours.

It is difficult for the system designer to furnish the system with a complete ontological model for interpreting sensor images. Even when an ontology is sufficient it need not necessarily be the most appropriate and transforming to a new ontology could be the only way to proceed. For example, circles are best described using the polar coordinate system ontology as opposed to the Cartesian ontology. Thus, there is a need for algorithms which adapt ontologies to system tasks and environments. Further, these algorithms should be equipped with sensor management strategies for selecting appropriate sensor configurations for the ontology and adapting signal processing software to interpret and classify the sensor data [4, 8].

This chapter is not concerned with hardware adaptation, for example active vision or sensor placement. Much work has been done in this field [4, 7, 13]. Some areas are discussed in this book. For a review of the different sensing strategies for visual object detection, object recognition and scene reconstruction, see [10]. Neither is this chapter concerned with the detection and recovery from various kinds of sensing faults including hardware failure and bandwidth limitations. Instead, this chapter emphasises the need for flexible signal processing software which is able to select the most appropriate signal processing tools for the task at hand, select model parameters and structure each tool appropriately and then determine explicitly an ontology for describing the behaviour of various multi-tool configurations in various environments. The signal processing software reconfiguration problem is similar to the sensor placement problem and it is not hard to believe that both share similar solutions including the syn-

---

2 An ontology is the set of entities presupposed by a theory.
thesis of analytical relationships, expert systems, model-based simulation and generate-and-test methods.

Figure 2: Aerial image segmentation of Edinburgh showing roughly sea, coast, suburbs and parks. The training stripes are shown on the left. On the right the areas around the stripes are shaded according to the region identified

This chapter develops information theoretic sensor software management strategies for infrared aerial image interpretation. The purpose is to determine an image ontology from unsegmented images and then configure sensors to discriminate the concepts within each image. The sensor system is supplied with a repertoire of simple generic signal processing tools. These tools could, in principle, be used in any sensing environment. Each tool determines some statistic of a local image region such as, for example, the average spectral energy or texture entropy. Tool configurations are then identified which exhibit the required image feature recognition coverage and also minimal redundancy. Our approach is able to determine alternative tool configurations in order to recover from software tool failure and it allows dynamic scheduling of the most informative tools by anticipating their informativeness using knowledge of the scene obtained from accumulating observations.

The next section introduces some simple signal processing tools for near-infrared, multi-spectral, aerial image interpretation. Section 3 explores the impact that sensor software reconfiguration can have on feature discrimination rates. Sections 4 and 5 introduce concept formation and signal processing tool selection algorithms. Finally, Section 6 describes a dynamic on-line sensor scheduling algorithm.
2 Simple Signal Processing Tools

This section describes some simple signal processing tools which are used to interpret aerial images obtained by a near-infrared, multi-spectral camera. This sensor records reflected radiation at a ground resolution of 20m in three, discrete wave bands - 0.50 to 0.59 microns (green band), 0.61 to 0.68 microns (red band) and 0.7 to 0.8 microns (very near infrared band). Fig. 2 shows the 512 × 512 grey scale rendering of a near infrared image of Edinburgh.\textsuperscript{3} We aim to construct an automated image segmentation and segment recognition algorithm. It is envisaged that the algorithm would be supplied with a large corpus of simple signal processing tools. The algorithm would then select a subset of these tools appropriate for its task. The task is to partition the image into spatial segments of roughly equal size so that when the sensor system traverses the same scene it is able to determine when a spatial segment boundary has been crossed. This is the localization problem.

Simple or complex signal processing tools are used which measure statistics (mean, mode, variance, entropy, histogram etc) of observables (luminosity, texture, shape, size etc). In the aerial image interpretation domain image statistics are determined from the reflected radiation energy $I(w)$ at pixel $w$. The statistics are evaluated over a sliding window of area 40 × 40 pixels. Each tool determines some statistic for $I(w)$ over subsets of the discrete spectral bands and outputs an ordering ($\succ$) of the various spectral bands - green (G), red (R), near-infrared (I) and non of these (B). The tool ordinal output is determined by some statistic $S$ which is calculated for each of the spectral bands $a, b \in \{G, R, I\}$. In general, $a \succ b$ whenever $S_a > S_b$.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{binary_images.png}
\caption{(a) Binary domination images from the SPOT near-infrared multispectral data of Edinburgh. Also, (b) binary domination sub-images and corresponding texture histograms.}
\end{figure}

Some tools operate on the binary spectral domination image $B(w)$. The\textsuperscript{3}This image was obtained from the European space agency satellite SPOT.
binary domination image for a spectral band is the set of pixels at which the intensity of the band exceeds that for all other bands. For all \( a \in \{ G, R, I \} \):

\[
B_a(w) = \begin{cases} 
1 & \iff (\forall b \in \{ G, R, I \}) \ 0 \geq I_b(w), \\
0 & \text{otherwise}. 
\end{cases}
\]  

The white areas in Fig. 3(a) depict the binary domination images for our image of Edinburgh.

**Luminosity Spatial Extent Tool (LD):** the luminosity spatial extent tool returns an ordering of \( \{ G, R, I, B \} \). A spectral band is preferred if it exhibits the greatest intensity over the majority of the pixel window. For \( a \in \{ G, R, I \} \) the tool determines the extent of domination of each band:

\[
S_a = |\{ w \in W : B_a(w) = 1 \}|. 
\]  

For \( a, b \in \{ G, R, I, B \} \):

\[
a \succ b \iff S_a > S_b. 
\]  

**Luminosity Variance Tool (LV):** the luminosity variance tool returns an ordering of \( \{ G, R, I \} \). A spectral band is preferred if the variance of its intensity value over the window \( W \) exceeds that of other bands. For \( a, b \in \{ G, R, I \} \):

\[
a \succ b \iff \sigma_{w \in W}(I_a(w)) > \sigma_{w \in W}(I_b(w)). 
\]  

**Luminosity Mean Tool (LM):** the luminosity mean tool returns an ordering of \( \{ G, R, I \} \). A spectral band is preferred if the mean luminosity value over the window \( W \) exceeds that of other bands. For \( a, b \in \{ G, R, I \} \):

\[
a \succ b \iff E_{w \in W}(I_a(w)) > E_{w \in W}(I_b(w)). 
\]  

Texture tools are used to evaluate the distribution of structure sizes within an image, including empty space (i.e. black) structures. The texture tools are based on Galloway’s *Primitive Length Encoding* [3] and operate on the binary domination images. The image is examined for neighbouring pixels of the same type (either filled or empty) which fit some structuring element - in this case a square of size \( d \) pixels. The number of image pixels \( N(d) \) with neighbourhoods that can contain the structuring element are counted for each \( d \in [1, 512] \). A histogram over values of \( d \), called the texture histogram, records the values \( V(d) = N(d) \times d^2 \).

**Texture Entropy Tool (TE):** the texture entropy tool returns an ordering of \( \{ G, R, I \} \). This tool operates over the individual spectral band texture histograms. The band with the greatest normalised histogram entropy is preferred. For \( a, b \in \{ G, R, I \} \):

\[
a \succ b \iff \text{Entropy}(V_a(d)) > \text{Entropy}(V_b(d)). 
\]
Texture Mode Tool (TM): the texture mode tool returns an ordering of \{G, R, I\}. This tool operates over the individual spectral band texture histograms. The band with the greatest histogram mode is preferred. For \(a, b \in \{G, R, I\}\):

\[
a \succ b \iff \arg\max_d(V_a(d)) > \arg\max_d(V_b(d)) .
\]  

(7)

To illustrate the operation of the five tools described above, Fig. 3(b) shows the binary domination images and texture histograms for a 40 × 40 spatial segment drawn from near the centre of the Edinburgh image at location (300, 300). The luminosity spatial extent tool yields 45%, 48% and 7% domination for each of infrared, red and green frequency bands. The band ordering is \(R \succ I \succ G \succ B\) which experimental data suggests could indicate a suburban area.

The qualitative sensor output is called a sensor cue and is a transfiguration of a set of quantitative observations made by the sensor. The sensor cue is a tuple comprising the tool descriptor, a representation of the observations and a label denoting the interpretation of the representation. So, for example, a sensor cue for the luminosity spatial extent tool output above, described in the language LISP, is:

\[
(:\text{tool} '\text{LD} :\text{rep} '(R I G B) :\text{interp} '\text{IR-R-G-B-ordering}) .
\]

3 Reconfigurable Sensors and Signal Processing Tools

Sensor selection is necessary when the number of sensors on platforms is prohibitively large for data processing [1]. Further, the preponderance of available signal processing tools requires that the sensors themselves are tailored to various tasks by selecting appropriate signal processing tools [5, 6]. The sensor system should reconfigure itself so that it is complete (i.e. apriori no objects are excluded) and this can be achieved in one of three ways: by learning when to reconfigure according to context; by recognising that a sensor (or tool) is not performing to specification or by reasoning about the problem domain.

Learning which sensors to use in various situations requires only basic statistical information obtained by observing the likely success rates of sensors. Such methods are generic, although they do require guaranteed complete experience of the problem domain. A more sophisticated statistical approach is to reason explicitly about each sensor but in such away that the reasoning is independent of the problem domain. Many sensors have generic (domain independent) performance signatures and it is possible to recognise when a tool is under performing. For example, some indication of the performance of the Kalman filter can be determined by continuously monitoring its innovation profile. Doyle’s SELMON system [1] uses a number of sensor-independent information measures to determine a sensor importance score - an importance ordering on the set of sensors. These scores, along with causal models, determine when and which sensor output is presented to the human system monitor. In contrast, Kastella [9] proposes a method for object discrimination which chooses which sensor to
apply based on the likelihood of an object given previous observations. The method chooses that sensor which maximises the Kullback-Leibler information (or cross-entropy) between the current estimate of the object probabilities and the expected estimate obtained from a sensor observation. In Hager’s task-directed sensor planning system for geometric sensing problems [4] both the discrimination gained and the cost of gathering and fusing information are figured into their method for selecting sensors and their placement. Deliberative methods require an understanding of the problem domain in order to anticipate the behaviour of sensors and their tools in unfamiliar situations. Deliberative systems are appropriate when insufficient prior experience is available to construct probabilistic models of the problem domain. We will not consider deliberative systems here (see [12]).

Our approach learns to recognise when to adapt tool sets using probabilistic models of the problem domain. There are two adaptation tasks for such a system. Firstly, models of the image concepts are adapted by (re-)clustering cues. Secondly, signal processing tools are reconfigured in order to distinguish these clusters. The concept formation and tool selection algorithms will be described in subsequent sections. The remainder of this section explores the relationship between features and their sensor descriptors and asks the question “How should cues be clustered and which features in an image singled out for optimal feature discrimination”?

In order to answer this question we need to consider multi-sensor systems and, for this purpose, introduce the idea of the aspect descriptor. A feature may look different when viewed in different orientations or under different weather conditions. An aspect descriptor is a multi-sensor description of a feature from one such vantage point. The complete set of all possible aspect descriptors for a feature constitutes a cue-centred model of the feature.

Definition 1 Assume access to sensors $S$ and signal processing tools $T$. Each sensor and tool is identified by a numeric index.

- A simple qualitative sensor cue from signal processing tool $t \in T$ is denoted $q_t$. All possible qualitative outputs from a sensor-tool $T(i)$ are collectively denoted $\mathcal{C}_{T(i)}$.
- An aspect descriptor $ss$ is an $n$-tuple formed from the qualitative output from $n$ signal processing tools $T \subseteq T (|T| = n)$: $ss \in \mathcal{C}_{T(1)} \times \ldots \times \mathcal{C}_{T(n)}$.
- A feature model $S$ is a set of aspect descriptors $S = \bigcup_i ss_i$.
- The function $\text{tools}(S)$ returns the signal processing tools which observe the cues in feature model $S$:

$$\text{tools}(S) = \{ t : q_t \in S \} .$$

An observed cue admits only a subset of all possible aspect descriptors. When cues are randomly distributed between aspect descriptors the number of
aspect descriptors admitted by \( n \) observed cues is an exponential decreasing function of \( n \) in general (see Fig. 4(a)). Fig. 4(b) shows the average number of feature models admitted by ensembles of aspect descriptors when the aspect descriptors are randomly clustered into feature models. Combining these two graphs yields the number of feature models admitted by various numbers of observed cues (see Fig. 4(c)) and it is plain to see that the number of admitted feature models undergoes a phase transition. The reason for this phase-effect is that cues, when shared between different admitted feature models, are un-informative. The phase transition occurs when the aspect descriptors with a preponderance of inter-segment shared cues are eventually filtered. The phase effect phenomenon is not restricted to randomly clustered feature models.

Figure 4: (1) Typical exponential decrease of the number of aspect descriptors admitted by increasing numbers of observed cues; (2) Average number of feature models represented by aspect descriptor ensembles of various sizes; (3) Average number of feature models admitted by different numbers of observed cues.

**Example** Binary sensors detect the presence or absence of a property and a binary aspect descriptor is a list of \( n \) truth values indicating the presence or absence of properties from \( n \) sensors. Suppose that all aspect descriptors have equal observation likelihood and no sensor is used more than once. From an
initial ensemble of \( N \) aspect descriptors, the number of descriptors consistent with \( t \) observations is:

\[
d_t = (N - 1)0.5^t + 1 .
\]  

(9)

Suppose that the aspect descriptors are initially assigned randomly to \( m \) feature models. An expression for the probability that \( n \) feature models are consistent with \( d \) randomly chosen aspect descriptors can be obtained iteratively:

\[
Pr_d(n) = \left[ 1 - \frac{n - 1}{m} \right] Pr_{d-1}(n - 1) + \frac{n}{m} Pr_{d-1}(n) .
\]  

(10)

Thus, the average number of feature models consistent with \( d \) aspect descriptors is:

\[
\bar{n}_d = \frac{1 - K^d}{1 - K}
\]  

(11)

where \( K = 1 - \frac{1}{m} \). Combining Eqs. 9 and 11, the number of feature models consistent with \( t \) observations is:

\[
\bar{n}_t = \frac{1 - K^{(N-1)0.5^t+1}}{1 - K} .
\]  

(12)

Naturally, for feature discrimination tasks the concept formation and tool selection algorithms should configure the system software to exploit the phase-phenomenon to the full. To do this the algorithms should endeavour to ensure maximal dissimilarity between feature models and maximal similarity within feature models. An optimal Bayesian concept formation algorithm is developed in the next section.

4 A Sensor-Centred Image Segmentation Algorithm

A segmentation is valuable only if the signal processing tools can discriminate different segments. Since the choice of tools is unknown by the human system designer a dynamic, sensor-centred segmentation method is required. But there is a problem. The most appropriate segmentation depends on the type of signal processing tools used and the most appropriate tools depend on the segmentation itself. To overcome this reflexive problem each tool votes on the most appropriate segmentation. The segmentation with the most votes wins. Of course, a segmentation which is heavily subscribed to will exhibit some redundancy in the signal processing tools. The segmentation algorithm developed in this section includes a method for filtering redundant tools.

There are two possible approaches to unsupervised clustering. The standard K-means approach clusters spatial segments directly based on neighbourhood and cue variation relationships. Alternatively, the sensor cues themselves can be
clustered and then spatial segments can be formed which correspond, in some way, with the highly auto-correlated cue clusters. The latter method is used here as it is more efficient: there are 142 different spatial segment types (i.e. aspect descriptors) to cluster in the Edinburgh image but only 39 different cues.

The clustering algorithm takes as input a joint probability distribution \( P_{r(z, z')} \) over the full range of observable cues \( z \) and \( z' \). This distribution is the expected likelihood that two sensor cues are observed together in an ensemble of local neighbourhoods within the image. The ensemble of local neighbourhoods is defined by a sliding window \( W \) which traverses the image along a number of stripes (our experiments use a window of size 40 \( \times \) 40 pixels and the training stripes shown in Fig. 2).

\[
P_{r(z, z')} = E_W P_{r(z, z' \mid W)} = E_W \left( \frac{n_z(W)n_{z'}(W)}{\sum_{z \in W,z' \in W} n_z(W)n_{z'}(W)} \right).
\]

where \( n_z(W) \) is the number of occurrences of cue \( z \) in window \( W \).

Cues are clustered in such a way that each cluster, \( G \) say, is as distinct from its background \( \neg G \) as possible. Using Bayes’ Rule the relative likelihood of \( G \) and \( \neg G \) given (conditionally independent) observations \( z_1 \) and \( z_2 \) is:

\[
\frac{P_r(G \mid z_1, z_2)}{P_r(\neg G \mid z_1, z_2)} = \frac{P_r(z_1 \mid G) P_r(z_2 \mid G)}{P_r(z_1 \mid \neg G) P_r(z_2 \mid \neg G)} P_r(G) \quad \text{Pr}(\neg G)\end{equation}

Optimally distinct clusters \( G_1 \) are those which maximise the expected relative likelihood of \( G \) and \( \neg G \) over all observations and clusters \( G \). When \( C \) is the set of all possible cue clusterings then:

\[
G_1 = \arg\max_{G \in C} E_{G \in G_1} \left( \log \frac{P_r(z \mid G)}{P_r(z \mid \neg G)} \right).
\]

The optimal clustering \( G_1 \) maximises the difference between the average cross-entropies of \( P_r(z \mid G) \) and \( P_r(z \mid \neg G) \) over all clusters and the average entropies of \( P_r(z \mid G) \) over all clusters. Thus, optimally distinct clusters maximise inter-cluster variation while minimising internal cluster variation as required by Section 3.

When extra constraints are applied to the segmentation task, such as image spatial extent limitations, the cue clusters can be manipulated further to conform to these constraints. The aerial image task specifically requires that segments are observed with approximately equal likelihood so that features are not too localised making them hard to find and also not too expansive making localisation difficult. The probability \( P_r(G) \) is an approximate measure of the likelihood of encountering the spatial segment associated with cluster \( G \). When \( N \) such segments are encountered with equal likelihood \(-\sum_G P_r(G) \log P_r(G)\) is maximised. Incorporating the spatial segment constraint into Eq. 16 gives:

\[
G_2 = \arg\max_{G \in C} E_{G \in G_2} \left( \log \frac{P_r(z \mid G)}{P_r(z \mid \neg G)} - K \sum_G P_r(G) \log P_r(G) \right) \log N.
\]
where $K \geq 0$ (in our experiments $K = 1$). Rewriting:

$$S_2 = \arg \max_{G \in \mathcal{G}} \sum_{G \in S, z} Pr(z, G) \log \left( \frac{Pr(G | z)[1 - Pr(G)]}{[1 - Pr(G | z)]Pr(G)^{1 + \frac{1}{\log N}}} \right)$$

(18)

where $Pr(G | z) = \frac{\sum_{z' \in G} Pr(z, z')}{\sum_{z'' \in G, z} Pr(z, z'')}$ and $Pr(G) = \sum_{z'' \in G, z} Pr(z, z'')$.

The image is segmented in one-to-one correspondence with the cue clusters. All pixels in the image assigned to a cue cluster are deemed to belong to the same spatial segment. A pixel is assigned to a cluster $G$ when, for some neighbourhood $W$ about the pixel:

$$G = \arg \max_{G \in \mathcal{G}} \sum_{z} Pr(z | W) \log \frac{Pr(z | G)}{Pr(z | \neg G)} .$$

(19)

Fig. 2 shows the result of unsupervised clustering of the Edinburgh image. The image is segmented into sea, dock area and city and includes the large scale park structures in the middle of the city.\(^4\)

Straightforward variations of the segmentation algorithm exist for when more control over the segment properties (e.g. number, composition) is required. For example, the algorithm can include a constraint fixing the number of clusters which can, in turn, be used to produce a hierarchical decomposition of the image by applying Eq. 16 or Eq. 18 recursively to the individual segments themselves. Further, when the image is (partly) partitioned apriori or multi-sensor associations exist between apriori segmented and unsegmented images, $Pr(z | S)$, the conditional probability of cue $z$ within some spatial segment $S$, can be determined by a window which covers each segment $S$ in turn. In which case, $Pr(z | S)$ replaces $Pr(z | G)$ in Eq. 19.

### 5 Signal Processing Tool Selection Strategies

Cue-clusters can be reduced in size because sensor cues may be ignored when they are either uninformative or they are redundant.

*Inter-cluster discrimination*: ignore cue $z$ which has a weak fidelity $\frac{Pr(G | z)}{Pr(G)} \approx 1$ with all clusters $G$.\(^5\)

$$\frac{Pr(G | z)}{Pr(G)} \approx 1 \Rightarrow Pr(z | G) \approx Pr(z | \neg G) \quad \text{[or } Pr(G) \approx 1 \text{]} .$$

(20)

*Intra-cluster cue redundancy*: ignore cue $z_1$ when it is significantly correlated

\(^4\)The raw quantitative sensor data is initially subjected to COMOC morphological filtering to reduce the number of different sensor cues.
with some other cue \( z_2 \) in the same cluster:\(^5\)

\[
\frac{\Pr(z_1, z_2)}{\Pr(z_1) \Pr(z_2)} \approx \min \left\{ \frac{1}{\Pr(z_1)}, \frac{1}{\Pr(z_2)} \right\}.
\] (21)

When all cues from a sensor are ignored then the sensor is redundant and can be deselected from the sensor configuration. Intra-cluster cue redundancy often offers many alternative ways in which the sensor configuration can be reduced in size. Denote as \( S_i \) the set of all possible reduced cue clusters for cluster \( i \) obtained by filtering redundant cues and filtering cues with weak fidelity. Members of \( S_i \) are called \textit{reduced cue clusters}.

**Definition 2** When \( a \) and \( b \) are two cue clusters, \( S_a \) and \( S_b \) are alternative reduced clusters for \( a \) and \( b \) respectively and \( t \) is a set of tools \( t \subseteq T \) then \( t \) is a \textit{sufficient tool suite} for distinguishing \( a \) and \( b \) when:

\[
\text{STS}(t, a, b) \iff (\exists s_a \in S_a, s_b \in S_b) t \supseteq \text{tools}(s_a) \cup \text{tools}(s_b).
\]

A sufficient tool suite \( t \) is a \textit{minimal tool suite} for distinguishing two clusters if there is no other sufficient tool suite that is a subset of \( t \).

**Definition 3** When \( a \) and \( b \) are two cue clusters, \( t \subseteq T \) and \( t' \subseteq T \) are sufficient tool suites, \( \text{STS}(t, a, b) \) and \( \text{STS}(t', a, b) \), then \( t \) is a \textit{minimal tool suite} when:

\[
\neg (\exists t') t' \subset t.
\]

Two heuristics are available to choose between possible reduced clusters. They are used when minimal adaptation of tools is required between clusters or when static, minimal but globally sufficient tool ensembles are required.

**Definition 4** A suite of tools \( t \) is a \textit{global tool suite} for cue clusters \( C \) if:

\[
(\forall a, b \in C) \text{STS}(t, a, b) .
\]

When \( t \) and \( t' \) are global tool suites then \( t \) is a \textit{minimal global tool suite} when:

\[
\neg (\exists t') t' \subset t .
\]

The \textit{minimal change tool suite} is the minimal set of tools which can be used to distinguish image segment transitions and also require minimal change between segments.

**Definition 5** When \( \tau = \{ "S_a \rightarrow S_b" \} \) is the set of all possible cluster transitions, \( t_{ab} \) and \( t'_{ab} \) are minimal tool suites for the transition \( S_a \rightarrow S_b \), then \( t \in \mathcal{T} \) is a minimal change tool suite if \( t = \bigcup_{a,b} t_{ab} \) and for all other tool suites \( t' = \bigcup_{a,b} t'_{ab} \):

\[
\sum_{S_a \rightarrow S_b} \sum_{s_i \in \tau, t_{ab}} |t_{ab} \setminus t_{bc}| + |t_{bc} \setminus t_{ab}| \leq \sum_{S_a \rightarrow S_b} \sum_{s_i \in \tau, t'_{ab}} |t'_{ab} \setminus t'_{bc}| + |t'_{bc} \setminus t'_{ab}|.
\]

\(^5\)The effect on \( \Pr \) by removing a sub-cluster \( C_2 \) from a cluster \( C = C_1 \cup C_2 \) is:

\[
\Pr(C \mid z) - \Pr(C_1 \mid z) = \Pr(C_2 \mid z)[1 - \Pr(C_1 \mid C_2)].
\]
To illustrate the efficacy of the reduced sensor suite, Fig. 5 shows a tri-
segmented image of Brighton. In part (a) the full range of signal processing tools
taken from section 2 (i.e. LD, LV, LM, TE and TM) are used to discriminate
the clusters. In part (b), a reduced set of tools (i.e. LD, TE and LM) exhibits
a near identical segmentation of the image to that of part (a).

Figure 5: Segmented image of Brighton, including segments 1 (city), 2
(South Downs) and 3 (English Channel): (a) full sensor suites for 1 =
\{LD, LV, LM, TE, TM\}, 2 = \{LD, TE, TM, LM\} and 3 = \{LD, TE, LV\} and
(b) reduced sensor suites for 1 = \{LD, TE, LM\}, 2 = \{LD, TE, LM\} and
3 = \{LD\}.

6 Dynamic Signal Processing Tool Scheduling

As observations accrue segment likelihoods change and segment hypotheses can
be filtered simply by the fact that they are extremely unlikely given the ob-
servations to date. The remaining possible segment hypotheses can be ranked
in order of likelihood. Such a ranking can be used to anticipate and select the
most informative signal processing tools from the minimum tool suite. However,
simply using the most informative sensor is often not enough as it is necessary
to ensure that all likely clusters are observable.

Suppose that observations $Z$ have been obtained by any sensor configurations
and correspond to some (as yet) unknown cluster $G$. The information $I_S$ that
is expected from $n$ new sensor observations $\{z_1, \ldots, z_n\}$ from a specific sensor
configuration $S$ of $n$ sensors $\{s_1, \ldots, s_n\}$ is:

$$I_S = \sum_{z_1, \ldots, z_n, G} Pr(G | Z) Pr(z_1, \ldots, z_n | G, s_1, \ldots, s_n) \log \frac{Pr(z_1, \ldots, z_n | G)}{Pr(z_1, \ldots, z_n | \neg G)}.$$
Assuming \( \{z_1, \ldots, z_n\} \) are conditionally independent then:

\[
Pr(z_1, \ldots, z_n \mid G, s_1, \ldots, s_n) = \prod_{i=1}^{n} Pr(z_i \mid G, s_i) = \prod_{i=1}^{n} \frac{Pr(z_i \mid G)}{Pr(s_i)}
\]

and:

\[
Pr(z_1, \ldots, z_n \mid G) = \prod_{j=1}^{n} Pr(z_j \mid G) .
\]

Thus:

\[
I_S = \sum_{z_1, \ldots, z_n, G} Pr(G \mid Z) \left[ \prod_{i=1}^{n} \frac{Pr(z_i \mid G)}{Pr(s_i)} \right] \left[ \sum_{j=1}^{n} \log \frac{Pr(z_j \mid G)}{Pr(z_j \mid \neg G)} \right]
\]

It can then be shown that:

\[
I_S = \sum_{G} Pr(G \mid Z) \left[ \sum_{i=1}^{n} \sum_{z_i} Pr(z_i \mid G) \log \frac{Pr(z_i \mid G)}{Pr(z_i \mid \neg G)} \right] .
\]

The value of \( Pr(z_i \mid G) \) can be obtained from the joint probabilities \( Pr(z_i, z_j) \) provided \( Pr(s_i) \) is the rate at which sensor \( i \) is sampled. When \( n \) sensors are used an equal amount to gather the training data for \( Pr(z_i, z_j) \) then \( Pr(s_i) = \frac{1}{n} \).

The most informative sensor suite \( S \) satisfies:

\[
S = \arg\max_S \left\{ \frac{I_S}{n_S} \right\}
\]

where \( n_S \) is the number of sensors in the sensor suite. The denominator is included as sensor suites may vary in size and, in such cases, it is prudent to maximise the expected information offered by each observation.

The efficacy of sensor scheduling for feature discrimination is illustrated by Fig. 6. Each trajectory in Fig. 2(b) is traversed by two sensing systems; the first employs the scheduling algorithm described above which selects tools with the greatest expected utility while the second algorithm chooses a random tool reconfiguration (each time). In both cases low probability cue clusters are filtered and the tools are chosen from minimal global tool suites for the remaining clusters. Within each spatial segment along a trajectory the selected tools measure observations \( z \) and the average value of \( \sum \log \frac{Pr(G \mid z)}{Pr(\neg G \mid z)} \) for these tools is recorded. The sum \( \sum \) of these averaged values is recorded as more observations are made as the sensing systems traverse the spatial segment. Fig. 6 represents the expected value of \( \sum \) over all spatial segments and trajectories. The graphs show the “probabilistic” counterpart \( Pr(G) \) for \( E(\sum) \) where \( Pr(G) \) satisfies \( \log \frac{Pr(G)}{Pr(\neg G)} = E(\sum) \).
Figure 6: True-positive segment identification probabilities $Pr(G)$ after sampling “# locations” many locations of the spatial segment. Solid and dashed lines show the true-positive rates for the dynamic tool selection algorithm and the randomly chosen global minimum tool algorithm respectively.

7 Conclusions

Our research is aimed at developing a sensor management and monitoring system which is able to guide the reconfiguration and interpretation of sensors and their image processing tools. This chapter has developed information theoretic sensor management strategies for infrared aerial image interpretation. An image ontology is determined from apriori unsegmented images and then infrared sensors are configured for optimal discrimination of the concepts within each image. Image concept primitives are obtained by a clustering algorithm which clusters significantly correlated sensor cues to maintain maximum (Bayesian) discrimination between clusters. Tool configurations are then identified which exhibit the required image feature recognition coverage and also minimal redundancy. The approach allows dynamic scheduling of the most informative tools by anticipating their informativeness using knowledge of the scene obtained from accumulating observations.

References


