Jigsaw: the unsupervised construction of spatial representations

Mark W. Peters and Barry Drake

Artificial Intelligence Group
School of Computer Science and Engineering
University of New South Wales
Australia
{markpeters, bdrake}@cse.unsw.edu.au

UNSW-CSE-TR-0007
November 2000

THE UNIVERSITY OF
NEW SOUTH WALES

School of Computer Science and Engineering
University of New South Wales
Sydney 2052, Australia

Abstract

A fundamental assumption in machine vision is that the spatial arrangement of pixels is given. In challenging this assumption we have utilised a general relationship that exists between space and behaviour. This relationship presents itself as spatial redundancy, which other researchers have considered problematic. We present a mathematical model and empirical investigations into this relationship and develop an algorithm, JIGSAW, which uses it to build spatial representations. The philosophy underpinning JIGSAW takes signal behaviour, rather than position, as primary. JIGSAW is an unsupervised learning algorithm that is efficient in time and space and that makes minimal assumptions about its operating domain. This algorithm offers engineering potential, opportunities in the understanding of biological vision, and a contribution to the wider field of cognitive science.

1 Introduction

Machine vision systems, such as robots, exhibit many shortcomings in their dealings with physical space, despite the technical excellence of real-time video and very fast computation hardware. Even very primitive animals, on the other hand, demonstrate great competence in the same problem domain. This is a sobering observation, particularly when we consider that animals do it for themselves whilst robots have teams of PhD's to help them. It may be that some fundamental assumptions of machine vision need to be challenged before it rivals biological vision in performance.

We describe JIGSAW, a new approach to vision that sidesteps a number of problems that seem to be aggravated by our standard formulation of visual information, and which has important implications for biological vision modelling. For reasons given below, we feel that this is a significant new way of processing visual information.

Contemporary image processing and machine vision rely heavily on the standard digital image formulation given by Gonzales and Woods (1993) in which an image is described by a list of pixels in a notionally orthogonal matrix. In order to read the value of a pixel we give its position, in either coordinate or list position form, as an argument to an image function. This function then returns the pixel's value. Hence, the position of the pixel is indexical. Whilst it is generally acknowledged that this is just one formulation of visual information, it is rarely questioned.

By contrast, this research begins by claiming that it is the *behaviour* of the pixel that is primary, and its position should be determined from this, rather than determined *a priori*. From this perspective, the standard ordered image matrix is observed as:

- an unnecessary dependency for visual processes
- an artefact of vision systems which is occasionally deleterious to some visual processes
- a confounder that obscures the importance of self-organising processes found in biological vision, but not its artificial counterpart
- a prosthetic assumption that reduces the robustness and autonomy of an intelligent agent.

JIGSAW's essential, *functional* departure from standard formulations is that visual information is treated as a self-organising set of signals whose proximity to each other in a geometrical space is determined by the similarity of their respective behaviours. The key implication of this is that the geometry is free to reorganise itself as signal behaviour changes.

An analogy may be helpful. Imagine trying to understand an image produced by a fibre-optic endoscope whose fibres have been randomly rearranged at the distal end. Any point of light that would have been visible under ordinary circumstances is still visible in the output image, but it is no longer likely to be in the right position. The problem that must be solved (before any standard image processing problems can be addressed) is the recovery of the input structure. If we agree that we are unable to tamper with, or even to examine, the distal part of the fibre optic endoscope, we must attempt to invert its scrambling function by some other means. All we have available is the unintelligible scrambled picture of points of light. So, in JIGSAW, we compare the behaviour of pairs of points of light, and move their fibres either closer together or further apart, if they are respectively similar or dissimilar in behaviour. **Figure 1** shows two arrangements of a set of inputs, one before and one after processing by JIGSAW. For this demonstration, the images are produced by providing a standard test image (shown on right) as input.

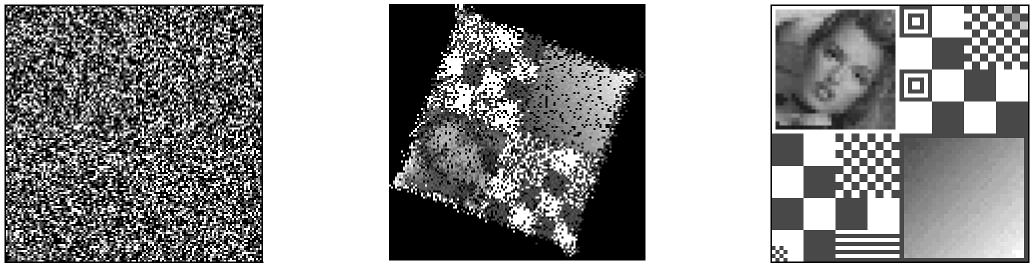


Figure 1: An initial geometry, a geometry processed by JIGSAW, and the original image “test128”.

When solving a jigsaw puzzle people use a number of heuristics. By analogy, the JIGSAW algorithm solves jigsaw puzzles in which no edge or corner pieces are identifiable from their shape, and no fit between pieces is exact. Under these circumstances, not all heuristics are available, so a puzzle solver would have to devote all their attention to the content of the picture fragments, and position them by exclusively using similarity heuristics. In a jigsaw puzzle, the elements have some spatial complexity, but in the domains we consider it is temporal complexity that enables the solution.

We believe that the principles under which JIGSAW operates are more general than the vision system that inspired it. In essence, JIGSAW is an algorithm for recovering the unknown spatial structure of a set of sensors. This recovery proceeds purely by considering the sensor readings, and without specific knowledge of what readings are expected at particular locations. This is achievable in principle when there is spatial redundancy in sensor signals. That is, if a number of sensors measure quantities influenced by a common region of space, then signal correlation across sensors can provide information on their spatial arrangement.

Spatial redundancy may have a more general meaning when dealing with non-physical spaces. Consequently, the problem may be stated even more broadly as one of finding a geometrical configuration for interpreting sensory information. The aim is to represent sensors within a space such that their geometrical relationship represents an actual relationship between the sensors, and where that relationship is manifest as redundancy (correlation) across the sensors.

The next two sections will give this research a context. In section 2 we explicitly state the objectives of the research which is followed by section 3 with the reasons we believe this endeavour is worthwhile.

Section 4 briefly introduces key terms and concepts used throughout the report, after which we present a number of sections discussing the theoretical background and empirical results of JIGSAW.

Section 5 provides the formal framework for aligning interpreted sensor arrangements with their actual positions. The usefulness of the algorithm presented in that section depends on a link between actual distance and behavioural difference. A mathematical model for this link is provided in section 6. Sections 7 and 8 report our empirical investigations into the theory covered in the two preceding sections. Section 9 takes a pragmatic look at JIGSAW and explores the enhancements and decisions to be made by a JIGSAW implementor. Section 10 reports our own experiences with JIGSAW in simulations and tests. Finally, section 11 discusses some general aspects of JIGSAW and points the way for future work.

2 Principles and objectives

This technical report is devoted to discussion of how to build spatial representations that do not already exist. As spatial representations are taken as given in much of vision research, we are obliged to explicitly state both why and how we do this. Our overarching policy is to use, wherever possible, a minimal, naïve algorithm, which assumes as little as possible. Our research philosophy is set out in more detail in what follows.

2.1 Autonomy-related goals

Peters (2000) presented a framework for viewing intelligence that recognised the essential contribution of *autonomy*. Using this framework, we have a number of autonomy-related goals.

2.1.1 Do not allow environmental knowledge to enter the calculations

It is an axiom of our approach that only sensors can provide information about the environment. Our understanding is that knowledge can come in three ways: by design, from others, and by discovery.

- We recognise that any useful operating principles imply a kind of inherent knowledge about the world, but we believe JIGSAW's inherent knowledge to be domain-independent, and therefore acceptable.
- JIGSAW does not receive communication from any other agent.
- All other JIGSAW knowledge comes from the sensors.

Unless information is contained in sensor signals, it must be considered inaccessible to the system. The reason for this is that we are trying to build a representation of the environment from something of a lower order. We cannot allow environmental knowledge into our representation-building process since that would defeat the purpose of the exercise. It would also make questionable the power of the algorithm and advance no real theories about primary visual processes.

2.1.2 Do not use global variables and ensure all functions operate locally

Though it is always possible to draw global knowledge from representations of the environment, we reject the option of giving this knowledge to the system. It is preferable that the system itself build its global knowledge from analysis of the data it organises. We have attempted to keep all operations local in order to avoid explosive computational complexity, retain the possibility of a credible biological interpretation, and permit operation of even the most naïve form of the algorithm.

2.1.3 Ensure that JIGSAW is a time-homogenous, anytime algorithm

Under JIGSAW, the state of the program is kept in the data. The algorithm should work no matter what the initial state of the data. Hence, any state can be considered a start point. The end point (halting condition) is not so freely identifiable, since all representations, no matter how small their theoretical error, should be considered provisional on there being no change in the sensor array. Since we do not assume a static sensor array, we cannot declare a point at which it is safe to consider the problem solved. Whilst the absence of a halting condition may be perceived as a drawback, we counter by proposing JIGSAW as a fundamental process, one which enables many other visual processes. It has a role analogous to that of a computer's operating system in that halting is an undesirable result.

2.1.4 Limit the complexity of the algorithm to O(n)

As with all algorithm design there is a desire to make it as efficient as possible, both in its processing time (time complexity) and memory requirements (space complexity).

JIGSAW is a non-terminating algorithm, therefore it is difficult to talk about its time complexity. However we see it as imperative to keep the time complexity of each processing chunk within order O(n) where n is the number of sensors. It would also be desirable to have a known, low order bound on the time to converge upon a good solution. However, such a notion is difficult to define and investigate scientifically.

There are no such ambiguities when discussing the space complexity of JIGSAW. As JIGSAW is required to store the represented locations for the sensors it is processing, the space complexity cannot be lower than order O(n). A goal in the algorithm design is to keep the space complexity at this order.

2.1.5 Reduce assumptions made about the space-behaviour relationship

JIGSAW should be able to work while making only the weakest of assumptions about the relation between space and sensor behaviour. The weakest, useful assumption is that space and behaviour are related by a monotonically increasing function, where space is ordered by distance, and behaviour by correlation. We prefer that JIGSAW work with a great range of functions relating space and behaviour, but note that not all versions of the algorithm will be able to do so.

2.2 Assumptions

Having stated that assumptions used by JIGSAW should be minimised, it is appropriate that we declare them.

In addition to the following three assumptions, we sometimes assume, to improve the results, that smaller behavioural differences are more reliable indicators of actual distances. This can be relaxed under some circumstances, so it is not a fundamental assumption.

2.2.1 The behaviour assumption

That behaviour is primary, and the behavioural difference between two sensor signals implies an actual distance between the sensors. We assume that there is a monotonically increasing relationship between the spatial-temporal separation of sensors and similarity of their signals.

2.2.2 The space assumption

That a space of a fixed, small dimension is sufficient to represent the distances between sensors. This assumption may be trivially satisfied for sensors in physical space, but may not be the case for abstract spaces.

2.2.3 The time assumption

That signals from the environment are temporally aligned so that meaningful comparisons can be made between sensors. Its important to state this assumption because it may not hold, for example, when batch processing data.

3 The importance of JIGSAW

There are both positive and negative reasons for considering JIGSAW as an alternative approach to existing methods of machine vision. The obvious negative is that JIGSAW is an additional complication with attendant processing requirements. We believe that the positives, given in this section, justify the cost. Firstly, JIGSAW offers great potential in a number of important application areas. Secondly, it avoids a number of problems inherent in standard methodologies of machine vision. Additionally, JIGSAW provides potential insight into biological vision.

3.1 Image processing and remote sensing

It will be apparent, as the reader becomes familiar with the details of this technical report, that JIGSAW is more than an algorithm. JIGSAW is a family of algorithms with modular components that are unified by a philosophy of the primacy of sensor behaviour and minimising domain assumptions. The modularity of the algorithm components and reduced assumptions make JIGSAW a useful engineering tool.

JIGSAW opens up interesting opportunities in remote data collection, auto-calibration, and robotics. Consider remotely sensing the seabed or planetary surface using a shower of sensors. JIGSAW can be used to recover the spatial arrangement of the sensors, despite their uncontrolled placement.

Furthermore, as JIGSAW is time-homogeneous, it can track sensors if they are slowly moving. The time-homogeneity of JIGSAW provides continuous calibration such that certain optical effects in vision systems are automatically compensated for. The JIGSAW principles enable it to merge multiple images, recover from changes in focal length, and provide compensation for distortions such as a cracked lens.

The modularity of the JIGSAW components means that as more domain information or more computational power is available, various enhancements can be included to manage trade-offs between conflicting philosophical goals of the algorithm.

3.1.1 Potential applications

We expect that JIGSAW will find ready application to machine vision and image processing tasks where there is (a) an intention to infer actual position in the world from position in an image, and (b) reasonable doubt about the reliability of the data in the image. Examples include vision systems that have become decalibrated for any reason (e.g., misalignment of axes, change of focal length).

Space-variant sampling produces inherent spatial distortion in images, which is difficult to invert without prior knowledge of the sampling function. JIGSAW promises to be able to perform this inversion autonomously, since it depends not on image position, but on signal properties.

Since it is a correlation-based algorithm, JIGSAW has potential to be useful in automatically separating environmental information from uncorrelated noise generated in the image transmission medium. Moreover, it may well have application in the field of data fusion, since it allows correlation, and correlation alone, to direct the association of signals. If there is sufficient correlation between, say, audio and visual signals, then the correlation will be expressed geometrically, possibly to the extent of fusing two data sources. This provides both a new approach to the old problem of data fusion, *and* a new definition of sensory modality. Under the principles of JIGSAW, signals are defined according to their correlation, not their supposed origin. A modality, therefore, is simply a set of signals that express a high degree of mutual correlation.

3.2 Problems in standard machine vision methodologies

There are four aspects of the standard formulation of visual information that can be seen as either problems or constraints. These were mentioned in section 1, and are now discussed in more detail.

3.2.1 An unnecessary dependency

The standard orthogonal matrix of pixels is not the only formulation for representing visual information. However, the matrix can be reconstructed if it does not exist *a priori* (Prokopowicz 1995), and all standard image processing methods are possible in its absence anyway (Bederson, Wallace and Schwartz 1995). There are therefore no binding reasons why it must be used.

3.2.2 An occasionally deleterious matrix

We show in section 8 that there are some inimical and inherent edge effects attendant upon the use of rectangular pixel matrices. Related effects are present in windows of all shapes. The fixed nature of the standard orthogonal matrix means it is unresponsive to spatial distortions, even those it creates itself. This property causes particular difficulty when we wish to fuse one rectangular matrix with another. Generally, one matrix has to give up its original orthogonality, and assume a spatial distortion consistent with the other matrix. Furthermore, all pixels in the rectangle are considered part of the data, even though this is often not the case (standard video, for example, is not exactly rectangular, since it has half-lines of data at the top and bottom of each frame).

3.2.3 A confounder that obscures important self-organising processes

To posit an orthogonal image processing domain is to assume a high level of *a priori* organisation in the vision apparatus. It is understandable that this is done so often, since it enables quick interfacing of machinery and techniques. However, it has a number of drawbacks:

- it must be ignored prior to a bottom-up assembly of similarity-based representations
- it hardens the percept-concept dichotomy by being a matrix within which concepts are sought, rather than part of a seamless multi-level representation system
- it cannot allow non-image data and image data to coalesce *automatically*
- it casts vision as a special category of problems rather than a subset of intelligence problem domains
- it does not help explain neuro-visual features such as retinotopicity, or axonal routing.

These constraints effectively fence machine vision off from other artificial intelligence research, and also limit what it can tell us about biological vision.

3.2.4 A prosthetic assumption that reduces the robustness and autonomy of an intelligent agent

Information that we, or other intelligent systems, process can come from outside, or be generated internally. The external information can come via other intelligent systems or directly from the world, and the internal information comes from some form of memory.

JIGSAW is relatively (as much as we can currently ensure) independent of other agents that tell it how the world is. Autonomy of this kind generally extends the range of operation. We should be more specific, and say that it is *autonomy of agency* that provides this enhancement to performance.

Unless we are content to deal only with systems that suffer from inherent dependencies on other agencies for the provision of pre-processed information, we must pay attention to autonomous self-organising processes of perception that enable some systems to dispense with other agencies. This is especially so if we are curious about just what processes those agencies themselves use to so cunningly arrive at the forms of information most easily digested by their dependents. Put in colloquial terms, perception is all about seeing for yourself, not being told.

JIGSAW works according to the principle of minimisation of operational assumptions. Assumptions are prefabricated decisions, made not on the qualities of the situation currently at hand, but on knowledge gleaned from other situations. They entail a form of learning by generalising from previous experiences. Learning systems that are built according to assumptions are circumscribed to the extent that they depend on those assumptions.

To make assumptions implies that one of the following decision-making processes has taken place. Either, (1) an intelligent system has made a decision about what is most probable; (2)

through a process of selection all those systems with a tendency to make the decision have survived and multiplied; (3) whoever created the system embedded certain behavioural predispositions that make the decision *de facto*; or (4) by pure coincidence the system behaves *as if* it has made the decision, even though no decision has been made.

To be independent of the first three forms of pre-fabricated decision-making entails a second equally valuable form of autonomy: *autonomy of assumption*. Autonomy of assumption allows a system to perform without the help of *a priori* information, regardless of its provenance.

For the most austere implementation of JIGSAW autonomy of agency is demonstrated by:

- an absence of domain-specific parameters,
- no provision of descriptions of the sensor array or transmission medium.

Autonomy of assumption is demonstrated by:

- no global knowledge collected or used by the algorithm,
- no working memory required, except for storing the represented geometry,
- independence from information about the starting or end points of problem solving,
- no organised *a priori* image data structures

3.3 Neuroscience

The processes by which retinotopicity and sensory homuncularity develop in the brain are still largely unknown (Kolb and Whishaw 1990). Neural connections in large brains like those of humans are believed to be genetically under-determined but adaptive in a number of respects. Brains are subject to hard physical constraints: they exist in tightly bound spaces, and there has to be a trade off between space devoted to neurons and the processes they extend to connect with each other (Mitchison 1991). Brains exhibit the ability to recover from injury and sensory tampering, which seems to indicate relatively high lability in axonal routing.

For each of the points above, JIGSAW offers interesting possibilities. JIGSAW creates the same kind of spatial ordering as the brain, using the same kind of information (signal content, not labelled lines). It builds representations that are concise, but information-rich, since the relationship between any two signals is expressed implicitly by their position within a geometry, rather than explicitly by an instantiated variable. Major reconfigurations of the input mechanism may initially make nonsense of any geometry JIGSAW develops, but it will respond immediately by developing a new geometry.

For these reasons, we believe that JIGSAW has a high degree of biological plausibility, and may be helpful in the area of mass-neuronal modelling.

4 Key concepts and terms

When trying to recover the spatial arrangement of sensors, the key that allows us to solve the problem is spatial redundancy. It has been shown empirically that vision systems, using a plane of pixel sensors, exhibit spatial redundancy. Some have considered such redundancy as a dispensable waste of transmission bandwidth (Kretzmer 1952) or system sensitivity (Srinivasan, Laughlin and Dubs 1982). However, this can only be considered a waste if the spatial arrangement is already known. Section 7.1 discusses this in detail.

Spatial redundancy presents itself as a relationship between, (a) the distance between two sensed points in the environment, and (b) the difference in behaviour of the sensor values. This relationship allows one, in principle, to make a good guess at the actual distance, a, between two sensed points based on the behavioural difference, b, between the sensor values.

Once an approximation has been found for a, the problem remains to find a geometrical interpretation for the sensors that reflects the value, a, for each pair. When a set of sensors are given interpreted geometrical positions, there will be an implied geometrical distance, g, for each pair of sensors. The goal then is to find a geometrical interpretation of the sensors such that $a = g$, for each possible pair.

We note that as this solution uses distances between pairs of sensed points, the resulting geometrical interpretation is invariant under translations, rotation and reflection. In fact, as parameters in the *ab* relationship are unknown, the solution is also scale-invariant.

For clarity, we reserve the term *actual distance*, a, to be the physical distance between the sensors in the world (or abstract analogue). We use the term *geometrical distance*, g, to mean the distance between the sensors as represented in their interpreted, geometrical positions. Finally, we use *behavioural difference*, b, to refer to the difference between sensor readings. Behavioural difference can refer to the difference between instantaneous, simultaneous readings or the average of this value over time. When important, this will be disambiguated explicitly or by context. Often the distinction is not important, so the term will be left ambiguous.

5 Formalising the search for geometry

The core of the JIGSAW algorithm works to minimise an error function that captures the difference between the actual positions of sensors and their geometrical representation. It proceeds by making changes to the geometrical representation to reduce this error, so that the geometrical representation corresponds to the sensors' actual positions.

For a pair of sensors, i and j, it is understood that the algorithm does not have direct knowledge of the actual distance between them, $a_{i,j}$. However it has been shown empirically that the difference in sensor value behaviours, $b_{i,j}$, is functionally dependent upon the actual distance between the sensors. The JIGSAW algorithm assumes that

$$b_{i,j} = f\left(a_{i,j}\right) \tag{1}$$

where f is an invertible function. As $b_{i,j}$ can be determined from the sensors themselves, $a_{i,j}$ is inferred using

$$a_{i,j} = f^{-1}\left(b_{i,j}\right) \tag{2}$$

The geometrical distance between two sensors is simply the Euclidean distance in the interpreted geometrical space. For example, for pixels in a two-dimensional space

$$g_{i,j} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2} \tag{3}$$

We define an error function, $\varepsilon_{i,j}$, which captures the error in geometrical distance with respect to actual distance. This is given as

$$\varepsilon_{i,j} = \left(g_{i,j} - a_{i,j}\right)^2 \tag{4}$$

For a sensor, i, the accumulated error in position is the sum of these component errors. This is given as

$$\varepsilon_i = \sum_{i \neq j} \varepsilon_{i,j} \tag{5}$$

and the total error for the system is the sum of the individual sensor errors. This is given as

$$\varepsilon = \sum_i \varepsilon_i = \sum_i \sum_{i \neq j} \varepsilon_{i,j} \tag{6}$$

The goal is to find some geometrical representation of sensors that minimises ε. If we consider hill-climbing algorithms, then the problem is to find some small change to the geometrical representation of the sensors which aims to reduce ε. The top-level algorithm is then,

1) Make a small change to the geometrical representation of sensors with the aim of reducing ε.
2) Go to 1.

As was mentioned in section 2.1.3, JIGSAW is a time-homogenous algorithm which does not terminate. This loop runs continuously to update the geometrical representation. In a batch processing version of JIGSAW various statistics may be used to terminate the loop.

We can clarify the top-level algorithm by considering the effect of moving only one sensor. It can be shown that if changing the geometrical representation of a single sensor, z, reduces ε_z, then the total error, ε, is reduced proportionally. As ε is the sum of individual $\varepsilon_{i,j}$, then moving z can only effect those components where i or j is z. We can partition ε into the three components, $\Sigma_{i\neq z}\ \varepsilon_{i,z}$, $\Sigma_{j\neq z}\ \varepsilon_{z,j}$ and $\Sigma_{i\neq z,\, j\neq z\ i\neq j}\ \varepsilon_{i,j}$. Moving z only affects the first two. As $\varepsilon_{i,j} = \varepsilon_{j,i}$, we can see that $\Sigma_{i\neq z}\ \varepsilon_{i,z} = \Sigma_{j\neq z}\ \varepsilon_{z,j}$, which by equation (5) equals ε_z. So the total error, ε, is changed in proportion to the change in ε_z.

Changing the geometrical representation of sensor i, to reduce ε_i, is guaranteed to reduce ε, so we have,

1) Select a sensor, i.
2) Make a small change to the geometrical representation of i with the aim of reducing ε_i.
3) Go to 1.

A particularly useful hill-climbing algorithm is the gradient descent method (Arfken, 1985, Morse & Feshbach, 1953). To move i using gradient descent, i is moved in the direction, $-d\varepsilon_i / d\boldsymbol{X}_i$, where $\boldsymbol{X}_i$ is a vector describing the geometrical position representing i. That is, we change the geometrical representation of i in the direction that will locally reduce the error.

We can factor the gradient descent into its components. Considering just the x component, let the change in geometrical representation of the sensor, i, be $h_{[x]i}$, then we have

$$h_{[x]i} = -k\frac{d\varepsilon_i}{dx_i} \tag{7}$$

where k is some small, positive constant. The parameter k controls the step size of the adjustments and is known as the descent rate or learning rate.

From the definition of ε_i, we have,

$$\begin{aligned} h_{[x]i} &= -k\sum_{j\neq i}\frac{d\varepsilon_{i,j}}{dx_i} \\ &= -k\sum_{j\neq i}\frac{d\left(g_{i,j}-a_{i,j}\right)^2}{dx_i} \\ &= -2k\sum_{j\neq i}\frac{\left(g_{i,j}-a_{i,j}\right)\left(x_i-x_j\right)}{g_{i,j}} \end{aligned} \tag{8}$$

As k is some arbitrary positive constant, we can simplify to,

$$h_{[x]i} = \sum_{j \neq i} h_{[x]i,j} \tag{9}$$

where

$$h_{[x]i,j} = k\left(x_j - x_i\right)\left(1 - \frac{a_{i,j}}{g_{i,j}}\right) \tag{10}$$

This says that for each sensor, j, we should move sensor i towards j, in proportion to its geometrical distance from j (by considering, x_j - x_i), but this is scaled by 1 - $a_{i,j}$ / $g_{i,j}$, which is negative when i and j are too close, and positive when they are too far apart.

A simplification makes the algorithm stochastic by sampling the population of possible pairs. This stochastic version considers a pair of sensors and adjusts their geometrical representation based only on their contribution to h. The stochastic, pair-based algorithm is then,

1) Select a pair of sensors, i and j.
2) For each dimension, x, in the geometry,
    - 2.1) Compute[1] $h_{[x]i,j} = k\ (x_j - x_i)\ (1 - a_{i,j} / g_{i,j})$
    - 2.2) Set x_i to $x_i + h_{[x]i,j}$
    - 2.3) Set x_j to $x_j - h_{[x]i,j}$
3) Go to 1.

The stochastic version of the algorithm provides benefits with the cost of randomness inherent in stochastic algorithms. The advantages include,

- smaller portions of work per iteration, with constant time complexity;
- the opportunity to draw the pair of sensors from a sample of the total population;
- the fact that the sample of pairs available may be created with a specific, useful bias.

The algorithm as presented requires a reliable estimation of $a_{i,j}$ as given by $f^{-1}(b_{i,j})$. The validity and reliability of this function is discussed in sections 6 and 7.

6 A mathematical model of the *ab* relationship

The key which allows us to extract spatial information from behavioural information is that spatial redundancy can be found in a set of sensor signals. Empirical evidence shows that the closer two sensors are, the more similar their signals are likely to be. This section examines the theoretical background to this phenomenon.

We can identify three sources of behavioural similarity. The first comes from the fact that sensors divide the world into regions of equivalent readings. While two sensors are in the same region, they will produce the same reading. The second source of similarity comes from the fact that sensors do not measure absolute points, but rather measure a small area, or set of points, in space-time. When a pair of sensors have overlapping sense regions, it is likely that the signals will have some similarity. The third source we have identified occurs by virtue of interactions in the world. The behaviour of two sensors may be similar because there is some common cause.

---

[1] The factor, $k(1 - a_{i,j} / g_{i,j})$, should be computed once outside the loop as setting x_i and x_j will modify $g_{i,j}$. This also makes the loop more efficient.

Let us consider two sensors with no signal correlation. Given two equivalent sensors, we can construct a notion of difference of readings. For example, if the sensors have binary states, then we may say that they have difference of zero when they are both in the same state, and difference of one otherwise. It is not important to this formulation how many states the sensors may have, merely that there is a metric that captures the difference of sensor readings, such that a sensor has zero difference with itself.

We can define a statistic, μ, which measures the expected difference between readings from a pair of sensors *where there is no signal correlation*. To calculate μ for a pair of sensors i and j, all that is required is the probability distribution of their values, $p_i(v)$ and $p_j(v)$ respectively. If the sensors produce discrete readings then

$$\mu = \sum_x \sum_y p_i(x) p_j(y) |x - y| \tag{11}$$

Using the example of a pair of binary sensors, where each state is equally likely, we have

$$\begin{aligned}\mu &= (0.5 \times 0.5 \times 0) + (0.5 \times 0.5 \times 1) + (0.5 \times 0.5 \times 0) + (0.5 \times 0.5 \times 1) \\ &= 0.5\end{aligned}$$

As another example, for sensors with equally likely readings from 0 to 255, we have

$$\begin{aligned}\mu &= \frac{1}{256^2} \sum_{i=0}^{255} \sum_{j=0}^{255} |i - j| \\ &= \frac{21845}{256} \\ &\approx 85.3\end{aligned}$$

For sensors that report real-number values, the equivalent formulation requires knowledge of the probability density functions, $p_i(v)$ and $p_j(v)$. In that case

$$\mu = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p_i(x) p_j(y) |x - y| \, dx \, dy \tag{12}$$

If in reality the signals from the sensors have some correlation, then the actual measure of behavioural difference, b, will be less than μ. If the signals are anti-correlated, then b will be greater than μ.

We now consider the effect that the distance, a, between the two sensors has on the estimation of expected difference.

Let the world be divided into regions such that if the sensors are in the same region, then the sensors have zero difference. Furthermore, assume that regions are randomly distributed in space such that region boundaries are encountered with an average frequency of λ per unit distance.

Given that the two sensors are separated by a distance, a, the probability that there are n region boundaries between them, P_n, is given by the Poisson distribution,

$$\mathrm{P}_n = \frac{(\lambda a)^n e^{-\lambda a}}{n!} \tag{13}$$

The probability that the two sensors are in the same region is the probability that there are no region boundaries between them, which is

$$\mathrm{P}_{n=0} = e^{-\lambda a} \tag{14}$$

The probability of straddling at least one region boundary is thus

$$P_{n\neq 0} = 1 - e^{-\lambda a} \tag{15}$$

When the sensors do not straddle a boundary, the expected difference is zero. When the sensors do straddle a boundary, the expected difference is μ. So the expected behavioural difference, b, as a function of a is

$$\begin{aligned} b &= 0 \times P_{n=0} + \mu \times P_{n\neq 0} \\ &= \mu\left(1 - e^{-\lambda a}\right) \end{aligned} \tag{16}$$

This formula provides a prediction about the relationship between the behavioural difference and actual distance between a pair of sensors. The parameters λ and μ scale the relationship.

The second source of behavioural similarity we have identified concerns sensors that have overlapping sense regions. A mathematical model of this effect is confounded by the complex ways in which sensors may aggregate the region. However, if we consider a pair of sensors, each of which takes a weighted sum of theoretical point-like readings, then the *ab* relationship is the weighted sum of the theoretical point-like relationships. This has the effect of averaging b over the region (in space and time). The larger this area of averaging, the more reliable is the *ab* relationship.

The final source of behavioural similarity comes from causal interactions in the world. Consider a ball on a table. The colour and shadow of the ball will depend on the characteristics of the light sources. The light from the table under the ball will be tinted by the colour of the ball. These complexities may lead us to ask whether a photo-detector is measuring light in its local vicinity, the reflectance of the object it is focused on, the brilliance of the light source or some complicated set of interactions.

We note that two points in space-time can be affected only by events in their past light-cones. The closer the two points are spatially and temporally, then the more their cones intersect, and the more chance there is that they are affected by common events. However, while this may be an interesting heuristic, causal relationships in the world are extremely complicated. For instance, they not only provide for correlating behaviour, but anti-correlating behaviour as well. We suspect it is unlikely that a general model will capture the effects of causality on the *ab* relationship.

To summarise, we recall that there is a statistic, μ, which is the expected difference between two sensor readings. The value μ is independent of any signal correlation and can be calculated from distributions using formula (11) or (12). We identified three sources that would make the measured behavioural difference, b, vary from μ. The classifying nature of sensors leads to the *ab* relationship given in formula (16). Overlapping sense regions could complicate the *ab* relationship, but we expect that sensors generally use simple aggregations over their sense regions. This will actually enhance the reliability of b values. Finally, there are causal interactions that may complicate the *ab* relationship for which we do not offer a general model.

Given the *ab* relationship (16), we can use the inverse formulation to provide an inferred distance, a', from measured behavioural difference, b,

$$a' = \frac{-\ln\left(1 - \frac{b}{\mu'}\right)}{\lambda'} \tag{17}$$

The success of this inference will largely be influenced by the assumptions underlying the formulation and knowledge of the parameters λ' and μ'. Parameter λ' is an estimate of the real λ, and parameter μ' is an estimate of the real μ. The sensitivity of the relationship to these parameters is explored in the next section.

6.1 Sensitivity to parameters

Using equation (17) it is possible to infer actual distance, a', from the behavioural difference, b. We now consider the effect on a' due to errors in setting the parameters λ' and μ'.

Shown in **Figure 2** is an idealised ab plot and a plot of a' versus b when μ is overestimated by 10%. It can be seen that for small values of b, a' is underestimated by a small amount. For larger values of b, a' is underestimated by a disproportionately larger amount.

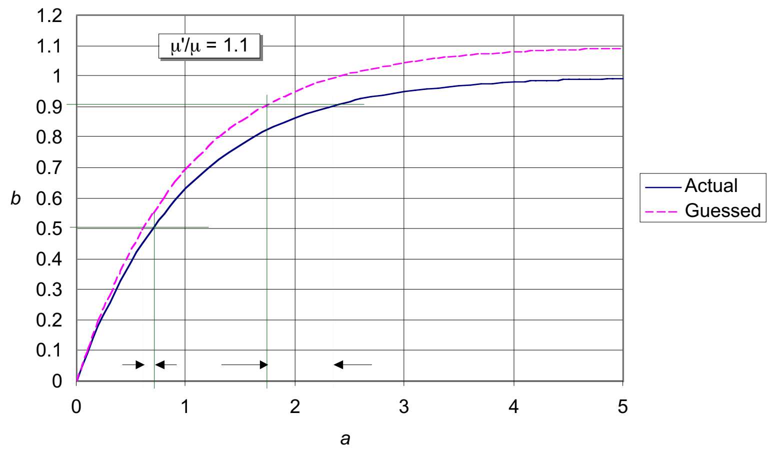


Figure 2: This shows a plot of the *ab* relationship and the effect of incorrectly setting the parameter μ' (here overestimated by 10%). When *b* is 0.5, *a* will be underestimated by a small amount. For a larger *b* value, *a* is underestimated by a disproportionately larger amount.

If the proportion of underestimation or overestimation of a' remains constant for all values of b, then a' will simply be proportional to a. As we are not interested in any effect that simply scales a' with respect to a, we will investigate the ratio, a' / a. If this is constant as λ' or μ' is varied, then the inferred actual distance is insensitive to that parameter (ignoring scale).

We can expand, a' / a, to give

$$\frac{a'}{a} = \frac{\lambda}{\lambda'} \frac{\ln\left(1 - \frac{b}{\mu'}\right)}{\ln\left(1 - \frac{b}{\mu}\right)} \tag{18}$$

Considering the system's sensitivity to λ', we see that varying λ' simply scales a' with respect to a. Consequently λ' can be ignored and set to some convenient constant.

To analyse the affect of μ', we define an error ratio as

$$\frac{\ln\left(1 - \frac{b}{\mu'}\right)}{\ln\left(1 - \frac{b}{\mu}\right)} \tag{19}$$

A plot of the error ratio versus b/μ shows how a' will be scaled for different measured behavioural differences. If this plot is a horizontal line then the scaling is constant and can be ignored. The more this plot varies from a horizontal line the more non-linear is a' with respect to a.

By considering this plot for different values of μ'/μ we can consider the sensitivity of the system to the parameter μ'.

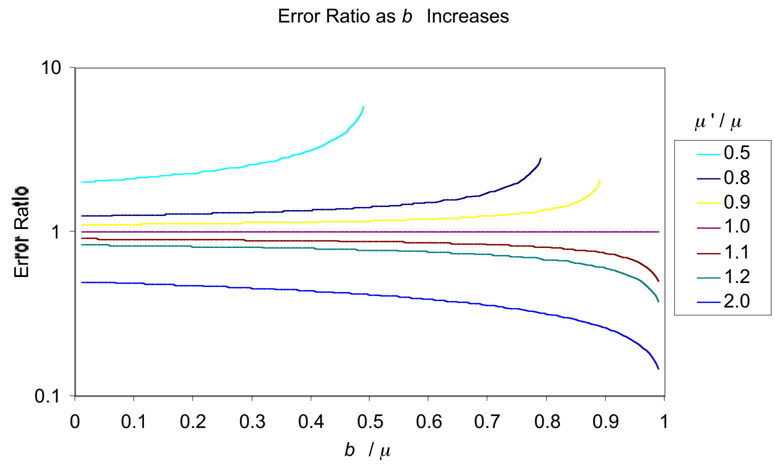


Figure 3: This shows the effect of over-setting or under-setting the parameter μ'. A horizontal line indicates perfect estimation. We see that when μ' is an over-estimation, then a' is an under-estimation. When μ' is an under-estimation, then a' is an over-estimation which rapidly increases as b grows. For values of b greater than μ' there is no estimate a'.

The plots in **Figure 3** show the error ratio with μ'/μ set to various values. The curves show that a' is only linear with respect to a when $\mu'/\mu = 1$ (i.e., μ has been correctly determined). When $\mu'/\mu < 1$ (i.e., μ has been underestimated), then the relationship between a' and a is highly non-linear. In particular, for values of b which are greater than μ' there is no value for a'.

The non-linearity can be seen more clearly when the curves are normalised to an arbitrary scale, as shown in **Figure 4**. The bottom curve in this plot shows the error ratio when assuming a very simple, linear ab relationship,

$$a' = kb \tag{20}$$

where k is some arbitrary scaling constant. This curve provides a bound on the error ratio when $\mu' > \mu$.

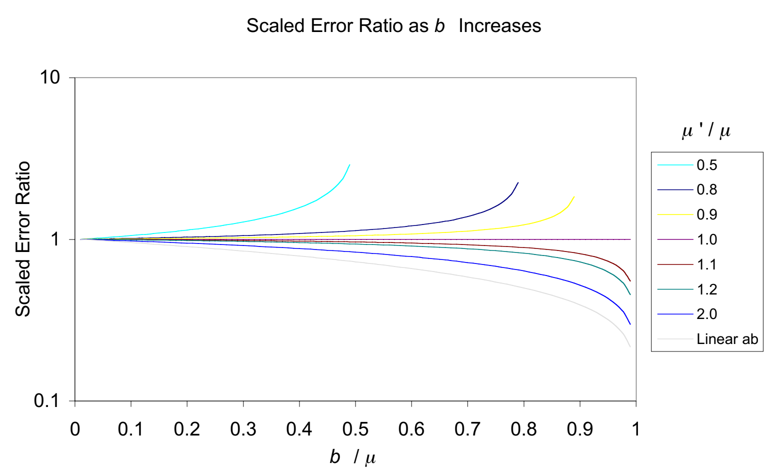


Figure 4: This shows the effect of over-setting or under-setting the parameter μ', ignoring any scaling factor, k. This highlights the severe effects of μ over-estimation. The bottom curve shows the error when assuming a linear relationship between b and a'. This provides a lower limit for under-estimating a.

When $\mu'/\mu > 1$ (i.e., μ has been overestimated), then the relationship between a' and a is slightly non-linear. The non-linearity is most pronounced for larger values of b where a' will be an under-estimation.

Considering that it is better to overestimate μ than to underestimate it, a conservative estimate may simply take the maximum possible difference between two sensor readings. This will never be an under-estimation and can never be worse than using a linear *ab* relationship.

We conclude this analysis by summarising key observations:

- The parameter λ' is unimportant and only scales the resulting approximation, a'.
- Reliable estimates of a are dependent upon good estimates of μ.
- It is better to overestimate μ (i.e., $\mu' > \mu$) than to underestimate it.
- With $\mu' > \mu$, small values of b give an approximately linear relationship between a' and a.
- With $\mu' > \mu$, large values of b give an under-estimation of a (i.e. $a' < a$), but this is no worse than assuming a linear relationship between a' and b.

Considering the algorithm's sensitivity to μ', we must address methods for obtaining good estimates. This is covered in section 9.

7 Empirical investigations of the *ab* relationship

In section 6 we developed a mathematical model relating behaviour to space. This relationship is at the theoretical heart of JIGSAW. To validate this model we report empirical results of investigations into the relationship.

7.1 Previous work

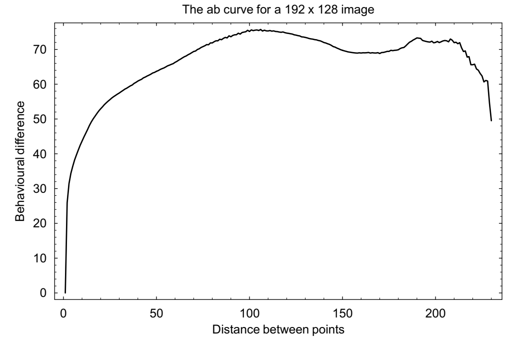
The empirical relationship between a and b was discussed by Kretzmer (1952). His measurements revealed the correlation between the distance that separates two points in television pictures and their similarity in luminance. He perceived this relationship as redundancy that might possibly be eliminated to increase the potential quantity of information contained in a TV signal of a given bandwidth. He was also aware that images of high local contrast (high λ, in our mathematical model) have much less redundancy. Higher redundancy, low λ, implies increased probability of predicting the value of one point given that of another. Srinivasan, Laughlin and Dubs (1982) confirmed these observations using ordinary photographs.

Prokopowicz (1996) recognised that pixels in close proximity have a high probability of sharing the same luminance, but treated this as a source of difficulty when searching for pixel matches across a short time interval. Indeed, such similarities are well known as the source of difficulty in optical flow and feature correspondence problems. In general, it can be said that the redundancy of images has often been perceived as a problem of one kind or another.

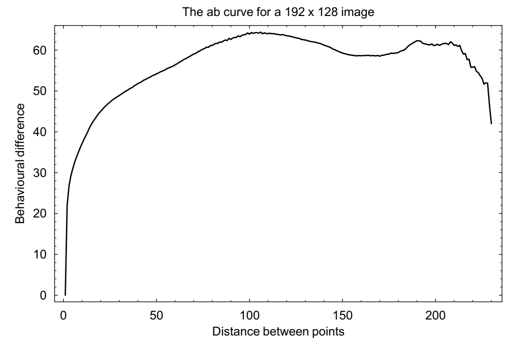
Peters (1998a, 2000) accumulated inter-pixel difference in luminance over time, using broadcast television as the input. Peters's results were consistent with those of Kretzmer and Srinivasan, Laughlin and Dubs, demonstrating, as might be expected, that the spatial *ab* relationship holds over time. As a result of using a long sample period he was able to extract relatively smooth and monotonic *ab* curves showing a high degree of redundancy. He also proposed our current formulation of the curve, and suggested that it may be present in all forms of data with spatial distribution. We pursue this suggestion later in this technical report.

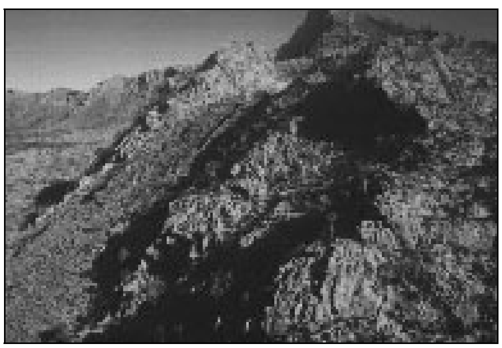
There seems to have been little general research into the *ab* relationship. This is probably because it is perceived as a merely incidental correlation between two kinds of information already known: luminance and position. We acknowledge that, to the extent that actual positions of sensors are known, the *ab* relationship can be superfluous. However, as so many machine vision problems stem from unreliable or unknown relationships between pixels and space, we conclude that the *ab* relationship is something that warrants considerable investigation.

Gosse Bluff Small – 192 × 128, range = 241

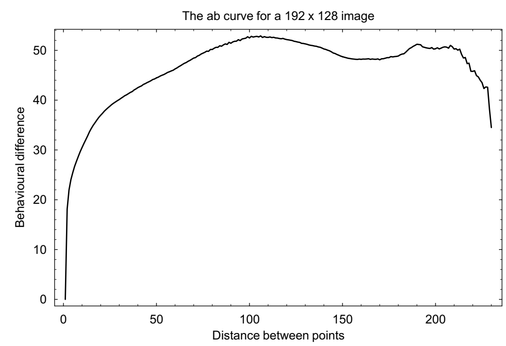


Gosse Bluff Small C1 – 192 × 128, range = 206





Gosse Bluff Small C2 – 192 × 128, range = 170



Gosse Bluff Small C3 – 192 × 128, range = 98

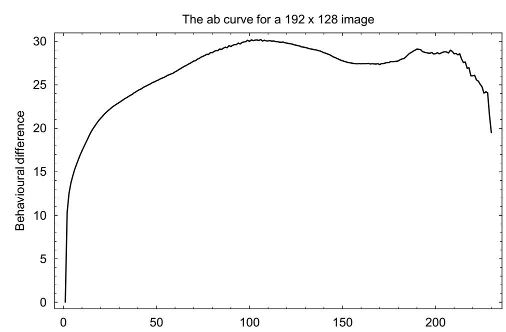


Figure 5: Changing image contrast affects μ in the *ab* relationship (see the vertical scale). No other effects are noticed.

We are not aware of any studies of how the luminance of a single pixel differs over a range of time intervals. In this experiment, we would be looking for a correlation between *b* and temporal distance, *t*.

To our knowledge, there have also been no previous studies of the *ab* curve for different classes of image. Yet it is clear from our research that regularities are present between image transformations and effects on the *ab* curve.

7.2 The effect of changes in *r*, λ and μ on *ab* plots

The primary variables, as has been shown in section 6, are λ and μ. To this we can add a third, *r*, the range of sensor values. In this section, we first discuss how image properties are related to these three variables, and then move on to other less universal features in the *ab* curve. All these variables and characteristics apply just as much to static images as they do to the sequences of images that JIGSAW is intended to work with. Consequently, and given the limitations of printed material, we use predominantly static images to show visual effects.

The ideal *ab* curve is governed by two parameters, λ and μ. Of these, μ is a measure of the average contrast, and governs the height at which the *ab* curve flattens out. When we reduce the contrast in a scene, we expect to see a proportional reduction in μ. **Figure 5** shows how the photographic effect of contrast reduction affects μ (the asymptote falls from ~80 to ~30) but not λ (the horizontal scale of the curve does not change). The dipping of the curve at high values of *a* is discussed in section 7.3.1.

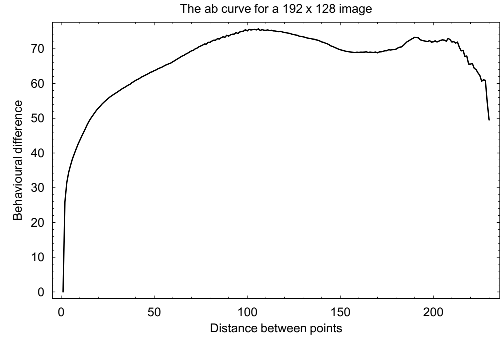
On the other hand, λ is a measure of the grain, detail, or sharpness of an image, and dictates how steeply the curve approaches the asymptote given by μ. More correctly, λ scales the *ab* curve horizontally, along the *a* axis, thereby compressing or extending the section of steepest gradient. **Figure**

1

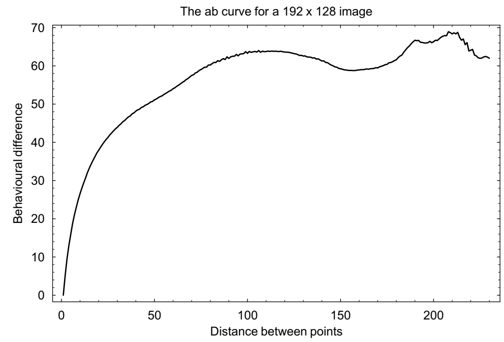
shows a photographic effect (Gaussian blurring) that affects λ. As the degree of blurring increases, the maximum height of the curve changes very little, but its initial gradient is progressively reduced as λ itself decreases.

Extreme blurring results in very low λ values and high correlation between actual distance and behavioural difference. Since low λ values can be obtained most readily from unfocused vision systems, it can be seen that an initial inability to focus an image may actually be conducive to recovering roughly localised positions. JIGSAW is unusual, particularly among calibration and orienting algorithms, in benefiting from such image degradation.

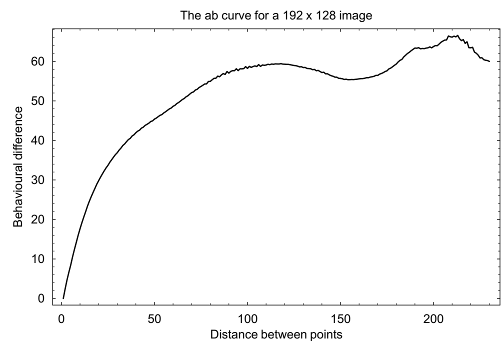
Reduction of *r*, the greyscale of a monochrome image, without reducing the contrast, has remarkably little effect on the shape of the *ab* curve. The extent of the greyscale should be thought of in a general way as the resolution of a sensor scale. In **Figure 7** we show three renditions of the same scene, the first with the original 220 greys and two modified images in which the number of greys is reduced to eight and two, respectively. The difference in chromatic resolution manifests itself as an exaggeration of the undulations of the *ab* curve, though the overall shape is hardly changed. This exaggeration is somewhat detrimental to JIGSAW, since it can only increase the possibility of error.

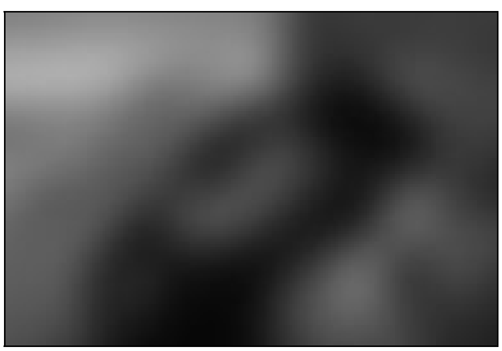
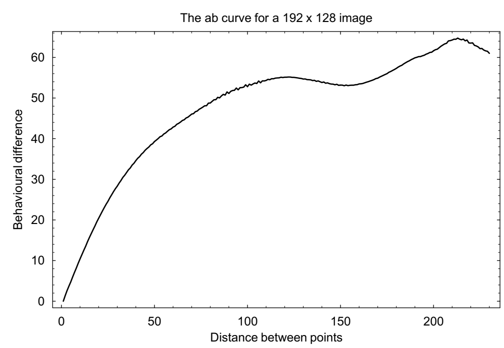
Gosse Bluff Small – 192 × 128, range = 241

Gosse Bluff Small B1 – 192 × 128, range = 217

Gosse Bluff Small B2 – 192 × 128, range = 196

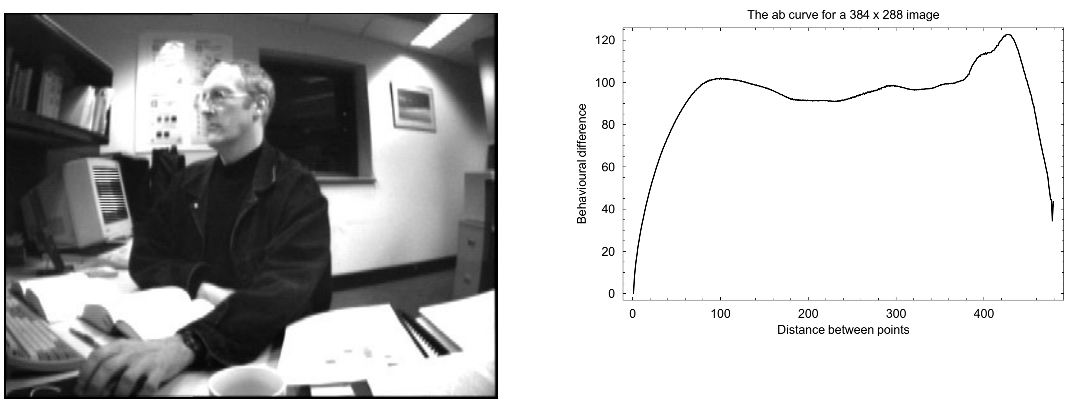



Gosse Bluff Small B3 – 192 × 128, range = 174

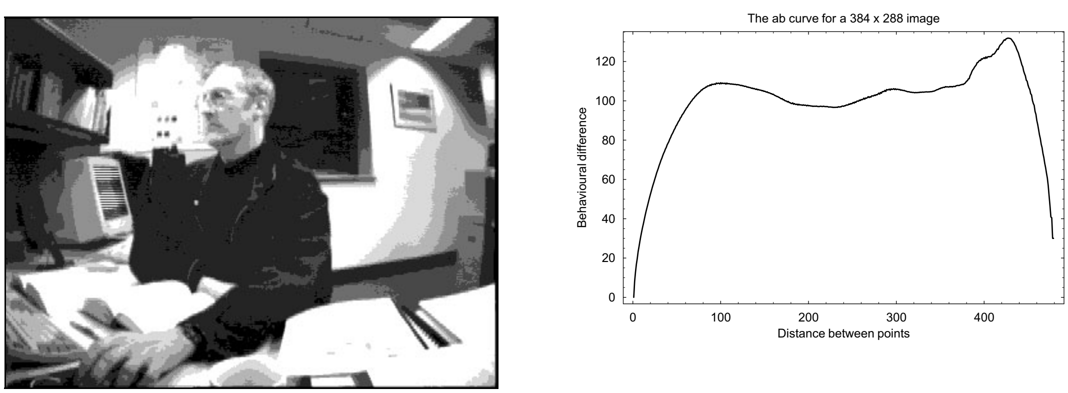
Figure 6: Blurring an image decreases λ and has a smoothing effect in the *ab* relationship.

In standard image processing, dithering is a method for reducing the number of colours, or greys, in an image. This consists of approximating an absent colour with an alternating pattern of two or more other colours. For instance, all greys can be approximated by a mix of black and white (e.g., see **Figure 8**). The salient effect of dithering is to drastically and disproportionately increase the average behavioural difference, *b*, measured over an actual distance, *a*, of one pixel – since the mixing involves the adjacent placement of different colours in an alternating pattern where before there may have been a single uniform colour. While the overall shape of the *ab* curve is relatively unaffected, it is no longer useful at small values of *a* (note the spike at the left end of the curve). As this part of the curve is often the most useful, dithering presents a major drawback. Fortunately, it is very rare in video signals.

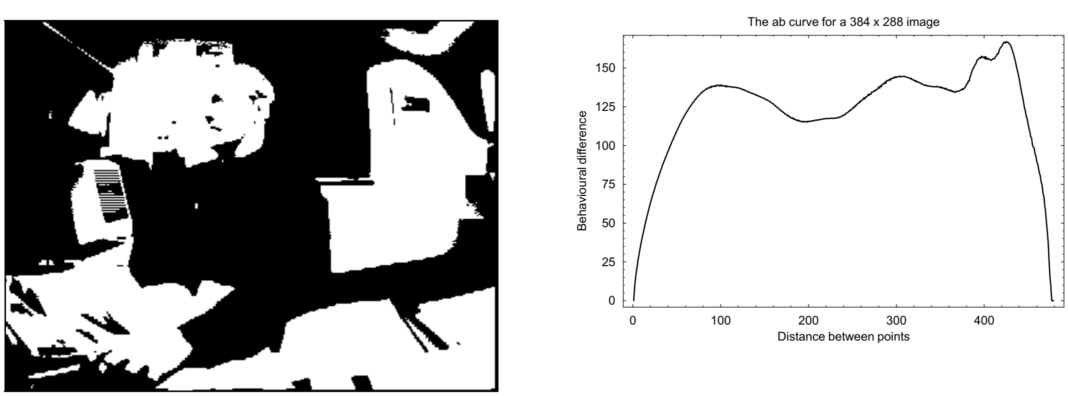
This section has discussed the general effects that image transformations have on λ and μ in the *ab* relationship. We now continue with a discussion of more specific observations linking image effects with the *ab* curve.



Office – 384 × 288, range = 220



Office 8 – 384 × 288, range = 8



Office 2 – 384 × 288, range = 2

Figure 7: Effect of reduction in resolution of sensor scale.

Office Dithered – 384 × 288, range = 2

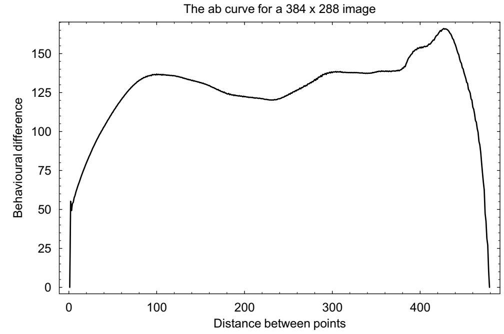


Figure 8: The inimical effect of dithering.

7.3 Variations of the *ab* curve

We have already seen, in **Figures 5 to 8**, dipping of the *ab* curve at high values of *a*. Under ideal circumstances the *ab* curve is smoothly monotonic, but we often encounter deviations from this form, and it is instructive to look at the relationship between these deviations and the visual effects that correlate with them.

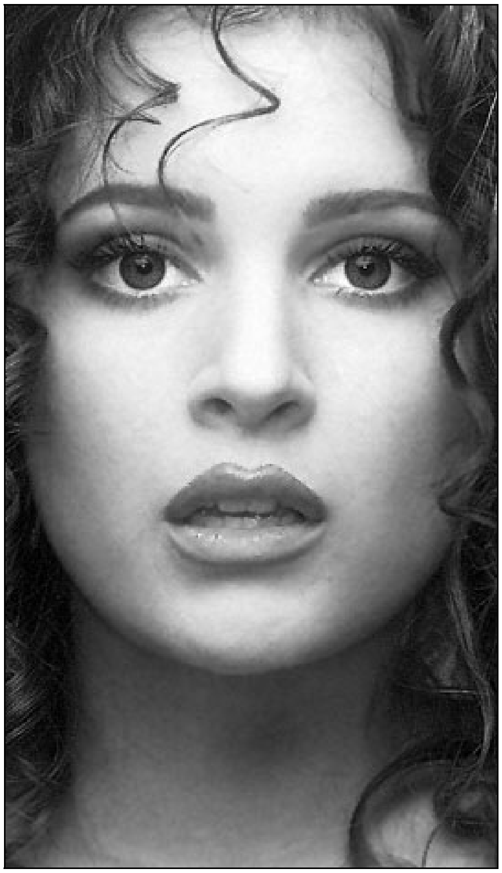
7.3.1 Smoothness

If the *ab* curve were linear, then JIGSAW would have no difficulty in inferring *a* from *b*. The non-linear nature of the *ab* relationship makes estimates of *a* susceptible to errors as *b* approaches μ. When λ is low, however, the approach to the asymptote is slow and the *ab* relationship is more stable. Consequently, JIGSAW works better with an *ab* curve of low λ, that is, whose initial gradient is low. Long, smooth gradients in an image result in a smooth *ab* curve. **Figure 9** illustrates this property. Here, λ is very low, and the *ab* curve extends across 200 pixels before beginning to deviate from its ideal form. This demonstrates that blurring is not necessary for low λ, it can occur naturally in some scenes. The *ab* curve in this image also demonstrates a common property of photographs: low *b* at high *a* values. We discuss this difficult phenomenon in detail in section 8.3.1.

Figure 9 shows a plot of the standard deviation from the mean for the *ab* relationship present in the image. The shape of the standard deviation plot is typical for the images presented here. This shows that small differences at small distances are highly probable. Great ranges of differences occur at middle distances. At extremely long distances, the data become very variable due to small population sizes (which also cause a corresponding decrease in the reliability of the *ab* curve). In this image, standard deviation is low at long distances simply because the corner areas are of similar tone. This phenomenon is discussed in section 8.4.2.

7.3.2 Spikiness

The image in **Figure 10** is a magic square, in which all values of grey from 0 to 255 appear, and the total quantity of grey in each row, column, and diagonal is the same. Furthermore, all quadrants share an average, as do each of their quadrants, and so on, down to the level of areas measuring 2 by 2 pixels. So, due to the special way this image is constructed, all small areas have an equal amount of light and dark, and the distribution of greys is completely uniform (256 greys in 256 pixels). The average behavioural difference is 85 – as predicted by formula (29). This spatial and chromatic consistency causes relatively low standard deviation in behavioural difference at mid-range distances, as can be seen in the standard deviation plot.



Jessica – 268 × 472, range = 254

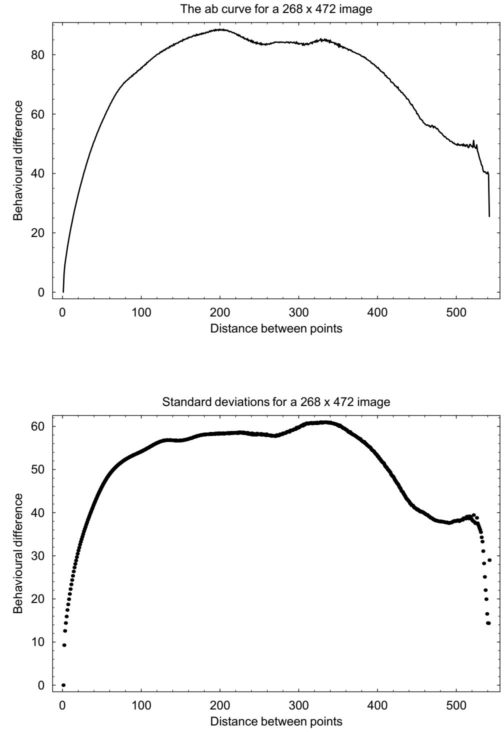
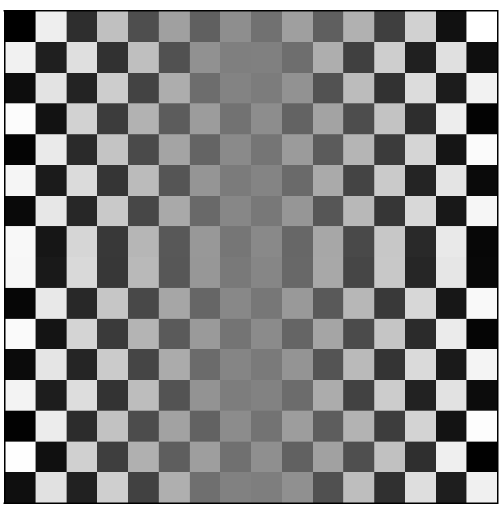


Figure 9: Long smooth gradients in an image produce a smooth gradient in the *ab* curve.



Magic Square - 16 × 16, range = 256

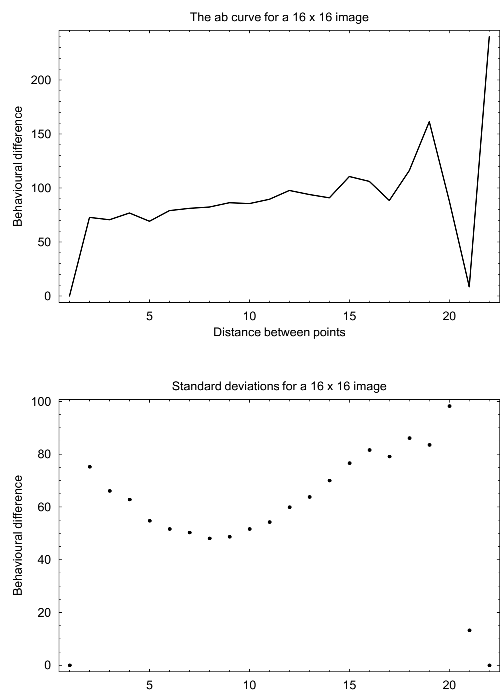


Figure 10: Standard deviation in an image of constant local average.

The spiky parts of the *ab* curve in **Figure 10** need to be explained (this is only partly due to the very small number of pixels). The extreme short and long distances in this image give consistently high behavioural differences This produces relatively high values for small and large values of *a* in the *ab* curve. Let us now look at the reasons for this. Nearly all distances of one pixel in this particular image have very high behavioural differences because of the chequerboard pattern. That accounts for the spike at $a = 1$ in the *ab* curve. At $a = 21$ the *ab* plot is measuring the differences over the longest distance in the image, which is its quantised diagonal. Inspection of the image itself reveals that across each diagonal there is again a very high difference in luminance (the actual difference is 240 – 0 = 240 for top left to bottom right, and 255 – 15 = 240 for top right to bottom left). At $a = 20$, one unit of distance less than the diagonal the behavioural difference is extremely small. This is because this part of the curve collects differences between one corner pixel and the two pixels adjacent to the opposite corner pixel. These differences are all very small, as can again be seen by inspecting the image. In short, the highly unnatural pattern of local contrast in this image causes an unusually spiky *ab* curve.

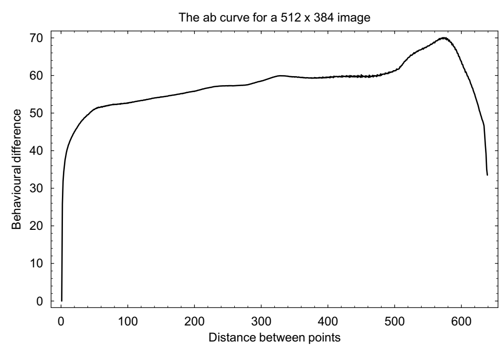
However, the standard deviation curve gives us a clue to another source of spikiness. As all local areas share the same average grey, over middle distances we should expect to get rather uniform behavioural differences, and this is what we do get. The standard deviation curve, however, shows a smooth section at mid-length distances. This, we know, is not due to reduced differences *per se*, so it must be due to an increase in population sizes. Later in this paper (section 8.4.2) we show that this is the case. In this image the actual distribution of distances is inversely related to the standard deviation of behavioural difference in this image.

7.4 The *ab* curve's immunity to spatial distortion

It is instructive to see what little effect spatial distortion has on the *ab* curve.

7.4.1 Proportionate scaling

Kosciusko – 512 × 384, range = 254



Kosciusko Small – 128 × 96, range = 249

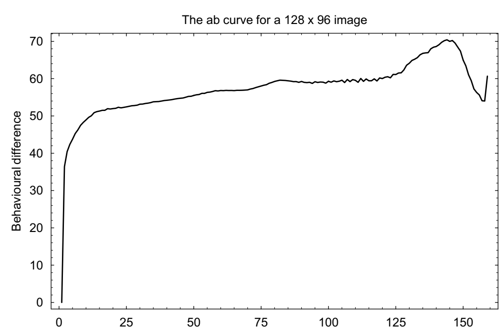


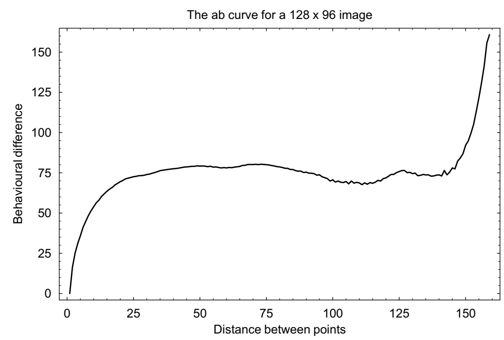
Figure 11: Change of resolution does not affect the *ab* curve.

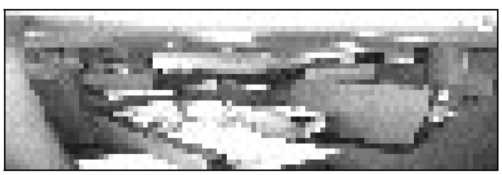
The effect (or the lack of it) of resolution change on the *ab* curve is shown in **Figure 11**. The aspect ratio of the source image is unchanged. Each dimension has been reduced by a factor of four, so the pixel population has been reduced by a factor of 16. Both images have a very high λ, each containing many small, high-contrast shapes. The only differences detectable in the *ab* curves result from higher variability in that of the lower resolution image, caused by a reduction in population size. This suggests that computation of an estimate of μ (i.e. μ') can be successfully accomplished using quite sparse sampling of the input population.

7.4.2 Disproportionate scaling

Figure 12 shows the effect of squashing an image in both dimensions, but by different amounts (from 128 to 100 pixels horizontally, from 96 to 30 pixels vertically). Both scale and aspect ratio have been changed. The *ab* curve is not greatly affected. The precise effect on the *ab* curve is quite hard to determine.

Lab – 128 × 96, range = 170





Lab Squashed – 100 × 30, range = 168

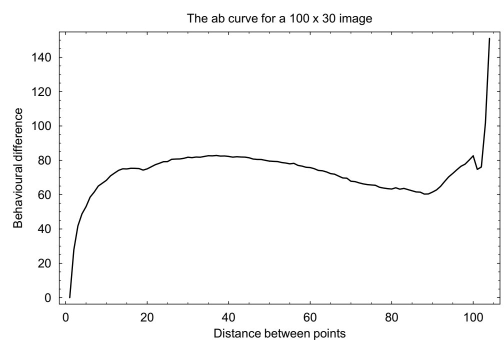


Figure 12: Resistance of the *ab* curve to disproportionate scaling.

7.5 Non-visual domains

The *ab* relationship was formulated in section 6 to be applicable more generally than to just visual domains, yet the discussion of *ab* curves so far has been restricted to images. In this section, we report on investigations into data collected by over 800 weather stations throughout Australia[2].

2 These data sets were obtained from the SILO website, http://www.bom.gov.au/silo/. The data are freely available, non-quality controlled observations. SILO is a collaborative project by the Bureau of Meteorology and the Department of Natural Resources, Queensland with assistance from Queensland Department of Primary Industries.

Figure 13 shows the average difference in total cloud cover measurements, taken at 9 am (local time) on 16 June 2000. It can be seen that the data follows the theoretical curve (shown in grey). The curve becomes unreliable for distances over 3,500 kilometres due to the low number of weather stations that far apart. We also note that the further apart the weather stations are, the more likely it is that they are in different time zones, which distorts the temporal alignment of the data.

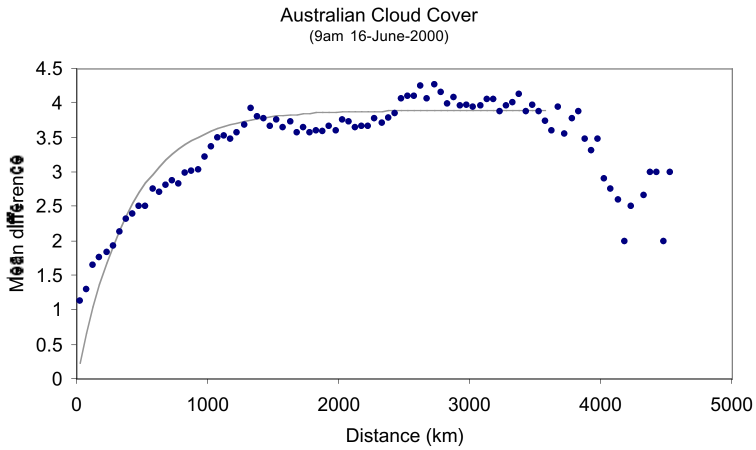


Figure 13: This chart shows the *ab* relationship for cloud cover data collected at weather stations throughout Australia. The curve becomes unreliable for distances over 3,500 kilometres due to the low number of weather stations that far apart.

Not all the metrics collected at the weather stations have such a good match with theory. **Figure 14** is produced in the same way as was **Figure 13** except that air temperature is plotted rather than cloud cover. It appears that temperature range and variation change significantly more from place to place than does cloud cover.

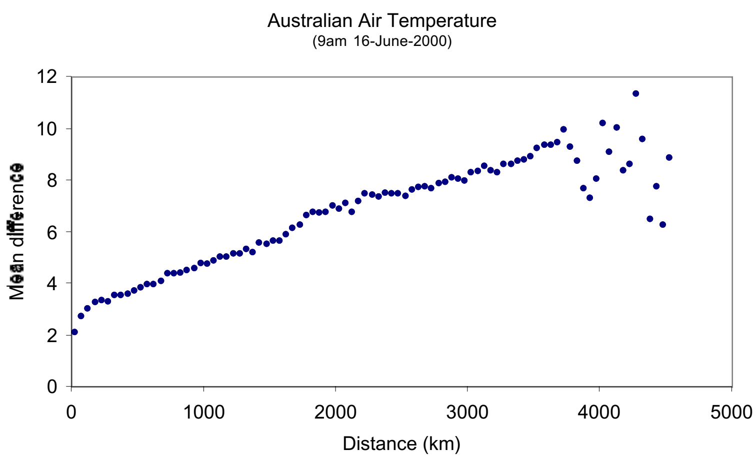


Figure 14: This chart shows the *ab* relationship for the 9 am air temperature (on 16 June 2000) for data collected at weather stations throughout Australia.

The explanation is that climate affects μ, as the distribution of sensor readings will vary from station to station. Here we see that μ is not globally consistent. The μ for pairs of stations in different climates will be higher than those of a similar climate.

When using time series data, it is possible to compensate for non-uniform μ by collecting sensor value distributions (see section 9.6). In **Figure 15**, the behavioural differences were measured over a period of 21 days. The charts on the left show the average behavioural difference for a given actual distance. To produce the charts on the right, μ was calculated for each possible pair by collecting the distribution of values at each station. For each pair of stations, b/μ provides a μ-corrected behavioural difference.

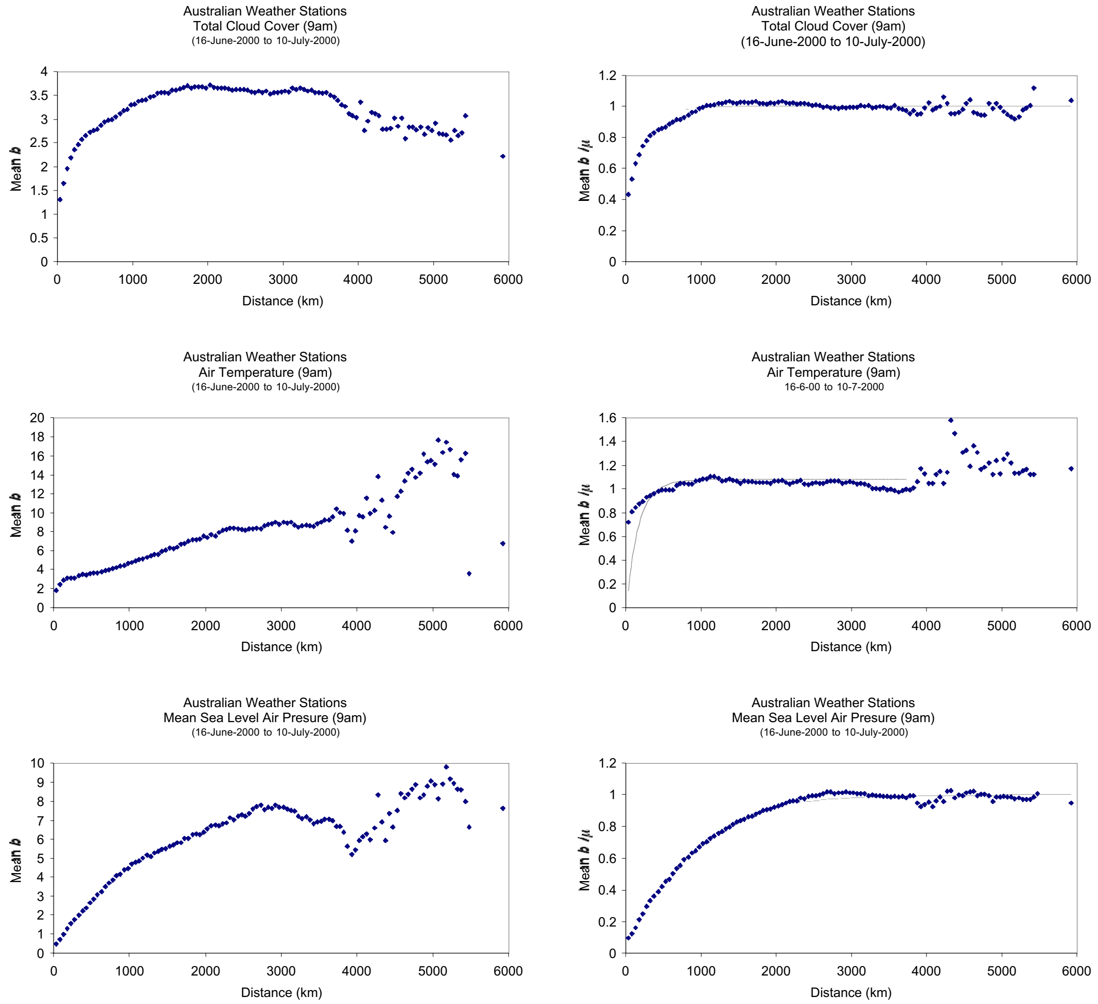


Figure 15: Correcting for non-uniform μ in weather data. The charts on the left show the average behavioural difference for a given distance. The charts on the right were produced by computing μ for each pair of stations and plotting the average b/μ for a given distance.

The Australian weather data provides encouraging evidence for the universality of the *ab* relationship.

7.6 Summary of the empirical investigations into the *ab* relationship

This section has examined the reliability of the *ab* relationship in static visual images and weather data. In general there is a reasonable match between the theory and data. For visual images, it can be seen that the *ab* relationship is robust under transformations such as change in contrast, sensor range, blurring and scaling. We saw in the weather data that local variation in the sensor reading distribution (climate) can be compensated for by using locally calculated μ.

Data transformations are one class of influence on the *ab* relationship. Other influences come from the nature of the sensor system. The effects of the sensor system are discussed in section 8.

8 Characteristics of the sensor system

Each sensor system will have its own characteristics, such as range, quantisation, noise, number and organisation of sensors, etc. In this section, we aim to give an outline of those characteristics pertaining to visual input, with occasional references to other domains. We discuss how characteristics were noticed, and why they are worthy of attention. We discuss sensors, the way pictures are framed, the subjects that are chosen, and how images are modified.

8.1 Sensor type

First, we will make a distinction between simple and complex sensors. A simple sensor is defined as having only one signal, or value over time. So far, most of our experimental work has been done with simple sensors.

A complex sensor will provide more than one signal value (or variable). When there is more than one signal per sensor they may or may not be reporting on a single location, so we may or may not need to assign the sensor a single geometrical position. Normally, however, a single location is all we are likely to consider.

Colour video experiments reported in section 10.2.3 involved complex sensors. The colour signals consisted of three values: red, green, and blue. In this case, we simply aggregated the differences. However, when sensors are of complex type, it is probably prudent to process each of their values independently. The behavioural differences in each variable can then contribute to the adjustment of the signal's geometrical position.

8.2 Modality

The modality of a sensor defines the physical quantity it measures. Modalities can be understood as the most basic of differences between sensory data. Our five human sense modalities are visual, olfactory, gustatory, auditory, and tactile. Within each of these, there are sub-modalities. For example, vision can be broken down into perceptions of motion, colour, form-with-colour, and dynamic forms, and it appears that specialised neural mappings exist for detecting each of these (Zeki 1992). If all our sensors are of the same type, we call this a single modality sensor array. To date, our research has concentrated on single modality sensor sets.

The possibility of setting JIGSAW the task of correlating sensors of more than one modality (e.g., sound and vision) begs the question of how modalities will be determined within the system. The answer may be that a set of signals will be considered a separate modality simply on the empirical basis of whether it is internally well correlated, but perhaps not particularly well correlated with other sets.

It may be desirable to invert, scale, or otherwise modulate signals of one modality, so that they match the response patterns of signals of another modality. From a pragmatic point of view there is no reason not to do this, but it assumes knowledge of the data and knowledge of the sensors that is in principle unknown.

It is hoped, given an arbitrary set of sensors, that where correlation exists the corresponding data fusion will be achieved, and that in cases where overall correlation does not exist, subsets of signals will simply congregate according to their local (modal) correlations.

8.3 Artistic deliberation

We have noticed a number of effects on the *ab* relationship which result from artistic deliberation. These effects are artefacts in that they result from decisions made by people, rather than properties of the natural world. This section categorises and discusses these effects.

8.3.1 Subject framing

Film and art directors, photographers and camera operators control the framing of images so that the subject of maximum interest and its most relevant surroundings are visible. The framing is obviously non-random. Though this is an impoverished description of their work, it is sufficient for the present discussion. The reason for discussing this subject is that the effect of human selection in imagery creates unusual properties that can reduce the ability of JIGSAW to organise data into useful spatial representations.

We note that recent moves to modify television standards have resulted in there being a number of different aspect ratios in concurrent use. This was always the case since wide-screen movies were shown on television, but now many commercials use a 2:1 or 16:9 aspect ratio too. On most televisions, all these various aspect ratios are displayed in a 4:3 format, occasionally leaving large blank areas at the top and bottom of television pictures. This makes use of television signals as a test domain for JIGSAW problematic.

8.3.1.1 Talking head effect

A subject of interest in human imagery is likely to be an active object rather than a static picture element. When a subject has been deliberately framed, it usually occupies a relatively central (often just off-centre) position in the image. It is rarely bisected by an edge of the image frame. The subject is also usually lit and coloured in such a way that it is easily separable from its background. Therefore, on three metrics (temporal, spatial, and chromatic) there is often more similarity between the edges of an image created deliberately than there is between the centre and the edge. This creates a marked deviation from the ideal *ab* curve, since over the longest distances (edge-to-edge, especially corner-to-corner) there can often be very little behavioural difference. Prolonged measurement of b is likely only to worsen the situation, since opposite edges will probably be much less active than the central area that separates them.

We call this deviation from the ideal *ab* curve the *talking head* effect, after the canonical example, in which, say, a newsreader's moving head and shoulders are framed against a relatively uniform and inanimate backdrop.

In **Figure 16** we demonstrate the talking head effect. In this photograph, the subject of interest has been deliberately framed within the bounds of the image, and is set against a contrasting uniform background. This results in an *ab* curve that returns almost to zero at its greatest extension. The standard deviations show that over very long distances there is consistently little difference between pixels.

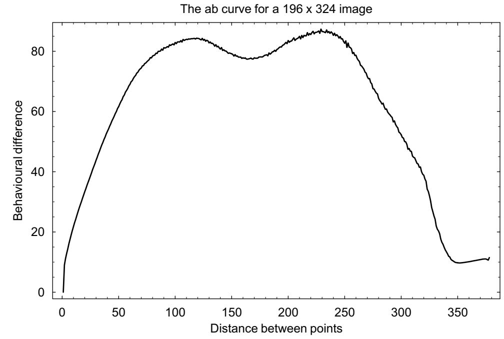
With the aim of quantifying the talking head effect, we conjecture that μ for pixel pairs near the edges of a window will be lower than for pairs towards the centre. To test for such a phenomenon, we start with a metric that attempts to capture the notion of a pixel being near the edge of the frame.

Let q_i measure how close a pixel, i, is to the edge of a frame with width w and height h. When i is on the edge of the frame we want q_i to be 1. When it is in the centre, we want q_i to be 0. For all other positions, we let q_i be a linear interpolation of these two extremes. This can be captured by the equation

$$q_i = \max\left(1 - 2\frac{x_i}{w},\ \ 2\frac{x_i}{w} - 1,\ \ 1 - 2\frac{y_i}{h},\ \ 2\frac{y_i}{h} - 1\right) \tag{21}$$

The *edgeness* of a pair of pixels, i and j, we define as the product, $q_i\, q_j$.

Bronte Small – 196 × 324, range = 152



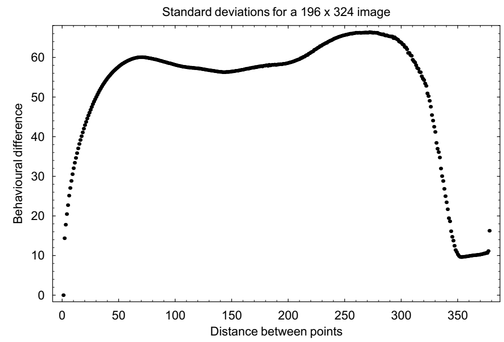


Figure 16: The talking head effect.

The charts in **Figures 17**, **19** and **20** show plots of μ versus edgeness for samples of 1,000 pixel pairs. The source scenes vary in how much talking head effect is present.

The source sequence for the plot in **Figure 17** consisted of approximately 15 minutes of video generated by a hand-held camera. The camera was randomly directed with no particular regard to artistic composition. We collected pixel value distributions for each pixel in the 128 × 96 window. Each μ value was computed from the pixel distributions (a sample of which is given in **Figure 18**). In **Figure 17**, μ is largely stable and uniform across space. The value of μ in this data set is a little lower than one third of the sensor range.

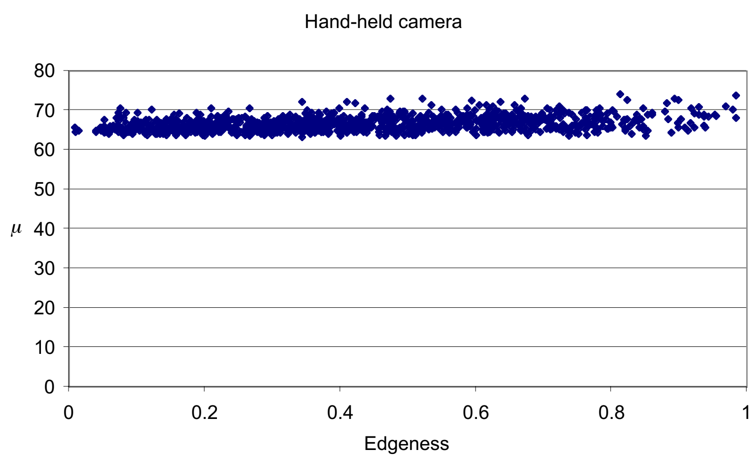


Figure 17: This is a plot of μ against edgeness for a sample of 1,000 pixel pairs. The source sequence is 15 minutes of video generated by a hand-held camera. The camera was directed with no particular regard to artistic composition.

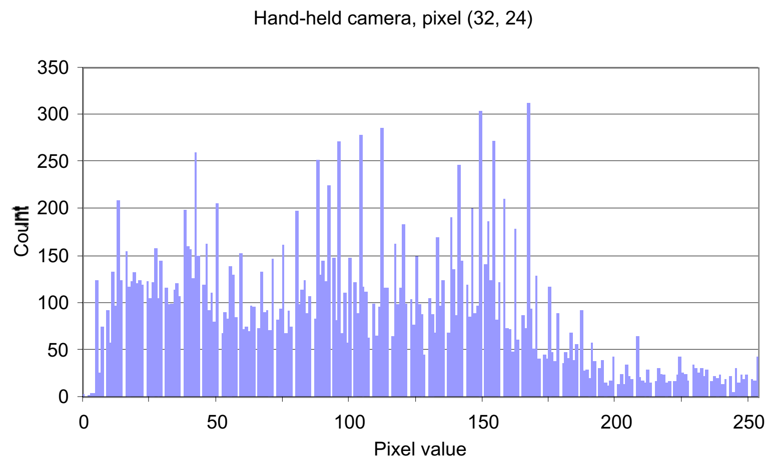


Figure 18: This is an example frequency plot for a pixel from the hand-held camera dataset. Note an artefact of the hardware leaves every seventh value unused.

The source data used for **Figure 19** consisted of 15 minutes of synthesised video input. Each frame in the sequence had a light coloured rectangle, of random size and location, on a dark background. Less than 10% of the rectangles were totally contained within the picture. This input simulates scenes with little talking head effect. We see that μ is largely independent of space, apart from a slight lowering of μ as edgeness increases.

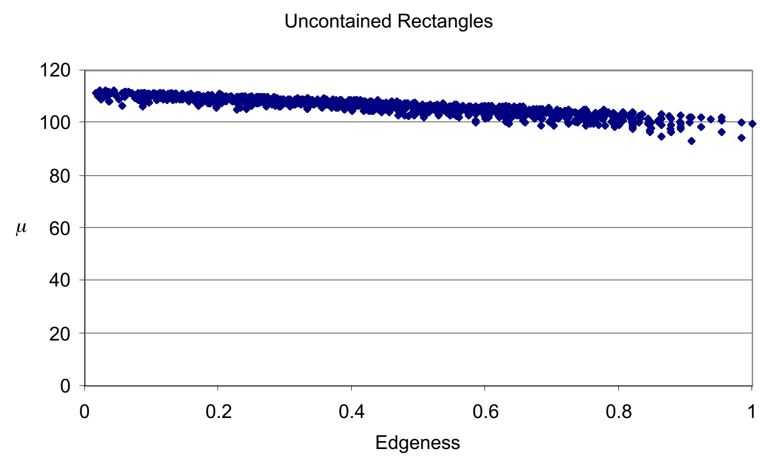


Figure 19: This is a plot of μ against edgeness for a sample of 1000 pixel pairs. The source scene is 15 minutes of synthesised video. The sequence consisted of light-coloured rectangles, randomly positioned and sized, on a dark background. Less than 10% of the rectangles were fully contained within the window.

Figure 20 is a dramatic demonstration of the talking head effect. The source scene is 15 minutes of synthesised video similar to the uncontained rectangles in **Figure 19**. The difference is that all the rectangles were completely contained within the window, to simulate talking head effect. It can be seen quite clearly that μ drops as edgeness increases.

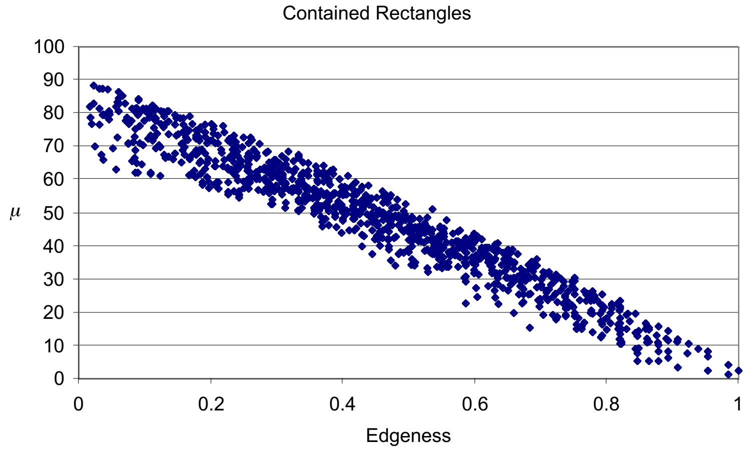


Figure 20: This is a plot of μ against edgeness for a sample of 1,000 pixel pairs. The source sequence was 15 minutes of synthesised video. The sequence consisted of randomly placed light-coloured rectangles on a dark background. All the rectangles were completely contained within the picture.

When μ is not spatially uniform, the impact on the *ab* relationship can be quite striking. We can see in the *ab* plot in **Figure 21** that the talking head effect has destroyed the *ab* relationship in the synthesised video using fully contained rectangles.

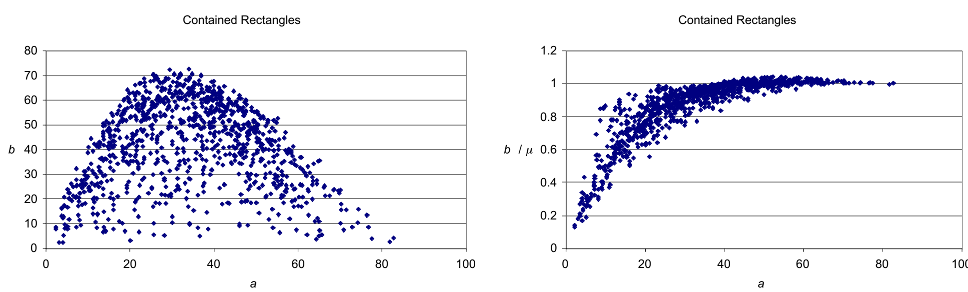


Figure 21: The left plot shows the *ab* curve for the contained rectangle sequence. The talking head effect has destroyed the reliability of the relationship: The right plot shows the dramatic improvement achieved using locally calculated μ, for a sensor system showing talking head effect. The *ab* relationship is transformed into a useable curve.

A spatially inconsistent μ can be corrected for by dividing b, for each pair, by the locally calculated μ for that pair. For the uncontained rectangle sequence, there is a small, helpful, but unnecessary improvement in the *ab* curve (**Figure 22**). However, for the contained rectangle sequence, the *ab* curve has been transformed from unusable to quite reliable (see **Figure 21**).

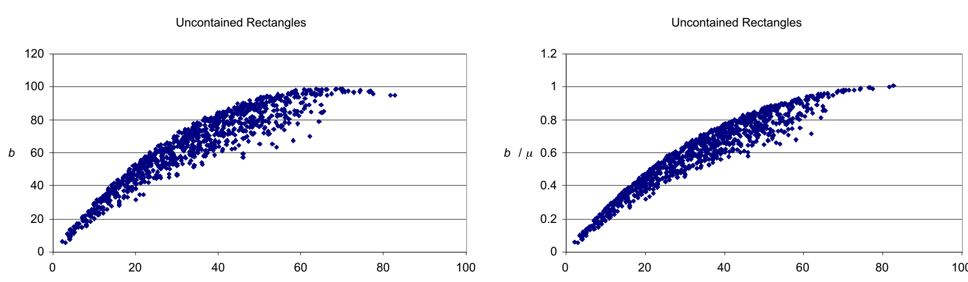


Figure 22: The left plot shows the *ab* curve for the uncontained rectangle sequence. The right plot shows the marginal improvement achieved using locally calculated μ.

8.3.2 Horizontal banding

Though we are usually unconscious of it, our everyday world consists of a horizontal band. Most activity takes place in this band and most information is located here. The horizontal band contains most of what interests us, and consequently it dominates our imagery and captivates our normal direction of gaze. The effect of all this is that many of our images have strong horizontal components. For example, many scenes consist of a band of sky, a band of buildings or trees and a band of ground (e.g., see the second sample image in **Figure 42**). This makes it easier to differentiate top pixels from bottom pixels than it is to distinguish left pixels from right pixels. This disparity is true of JIGSAW. It has the consequence that the first ordering of pixels performed by JIGSAW is usually a separation of horizontal bands.

8.3.3 Graphic elements

It is now common for television stations to overlay pictures with their logo or other persistent graphic elements. The persistence and spatial uniformity of these graphic elements can lure JIGSAW into over-condensing the pixels that carry them. A particularly deleterious example is a box used as a frame for the presentation of numerical sports results. If the frame is uniform in colour, many of the pixels that make it up will behave identically while it is visible. Small (or zero) behavioural differences over long actual distances over long time intervals cause spurious (but temporary) spatial ordering in JIGSAW.

8.4 Inescapable artefacts

In addition to the transmission effects that are deliberately introduced (and therefore theoretically avoidable) there are a few inescapable artefacts of digital imagery: quantisation, aspect ratio, and edge effect. This section will address these three. Quantisation has been shown to have negligible effect on the *ab* curve, but the same cannot be said of aspect ratio and edge effect.

In image processing, quantisation can be spatial, temporal and chromatic. We will discuss all three forms. Aspect ratio affects both the distribution of distances within an image, and the intensity of the edge effect in different orientations. Edge effect causes points within an image to have different distributions of available distances, so some points converge on their position less quickly than others.

Moreover, the distribution of distances in any image is not uniform. And, what is more, the distributions of distances are not uniform across an image, that is, they vary from point to point. These facts become important when randomly selecting pairs for processing – since parts of the *ab* will be preferred by virtue of the distribution and this preference may be sub-optimal.

8.4.1 Quantisation: spatial, temporal, and chromatic

In all digital image-forming equipment, sensors deliver quantised information. Spatially, the world is divided into pixels (e.g., 768 × 576), temporally it is divided into 25 (or 30) frames per second, and chromatically is it divided into 256 greys, or a range of colours. There are occasionally discrepancies between sampling frequencies. For example, the CCD in one of our cameras cannot achieve 768 × 576 resolution. The camera interpolates to create an analogue video signal of 576 lines, which our frame grabber resamples into 768 digital columns. The same camera can measure only 220 levels of grey, but the frame grabber encodes these using values from 0 to 255. Every seventh grey level in the 0-255 scale is unused (see **Figure 18**).

Quantisation of greyscale also creates edge effects – relatively higher contrast for greys that are at the extremes versus those in the middle of the scale. JIGSAW is able to handle these quantisation artefacts without significant degradation of performance (see **Figures 7**, **11**, and **18**).

8.4.2 Non-uniform distribution of actual distances in a window

As was explained in section 5, JIGSAW selects pairs of sensors and adjusts their geometrical representation based on *b* and *g*. Furthermore, it was shown in section 7 that the reliability of the *ab* curve is better over short distances compared to long distances. This prompts the following question, “What is the distribution of *a* for some set of sensors?” If *a* is naturally distributed to smaller values then this is beneficial to JIGSAW. In general, the distribution of *a* for a given set of sensors has an impact on the distribution of *b*, which in turn effects the performance of JIGSAW.

We consider a set of sensors as providing a window into the world. The distribution of *a* will depend upon the shape of the window and the distribution of sensors in the window. This section empirically explores the distribution of *a* for some obvious shapes.

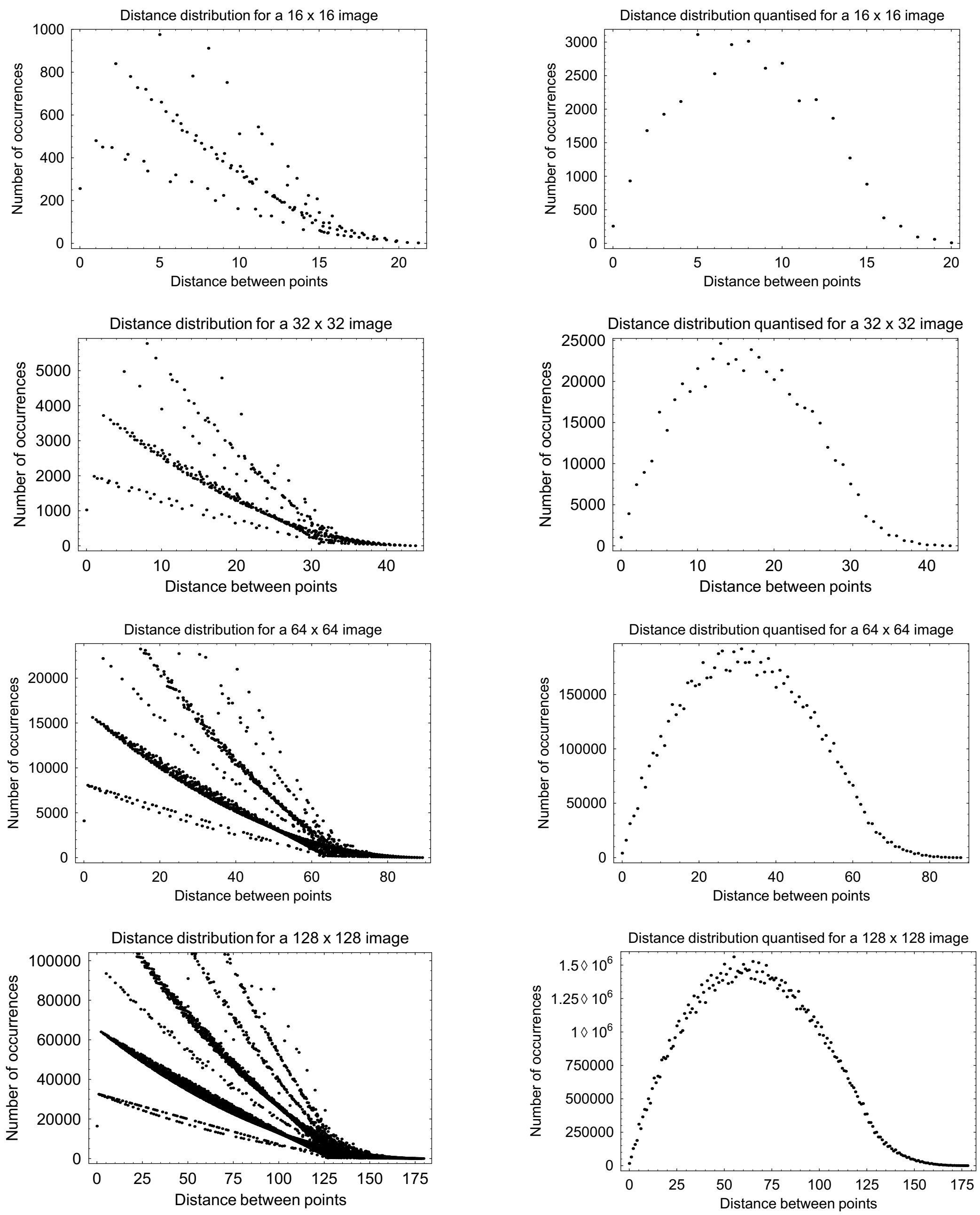


Figure 23: Scale invariance of actual distance distribution in rectangular windows.

8.4.2.1 Rectangular windows

We begin by looking at the distribution of *a* in a rectangular window, where the pixels are evenly distributed in a rectilinear grid. This is a common format for image processing and, as will be obvious from the graphs in **Figures 23-27**, the distributions are indeed non-uniform.

Between pairs of pixels in windows of common aspect ratio there are many more mid-length distances than there are either extremely short or extremely long ones. The shape of the distribution curve is dependent only on the aspect ratio of the image, not its size. In **Figure 23**, the left-hand charts plot frequencies (number of occurrences) of actual distances, the right-hand images contain the same data but integer quantised to make the distribution curves clearer. It can be seen that for a pair of pixels randomly selected from a window of common aspect ratio there is a high probability of *a* being of mid-length.

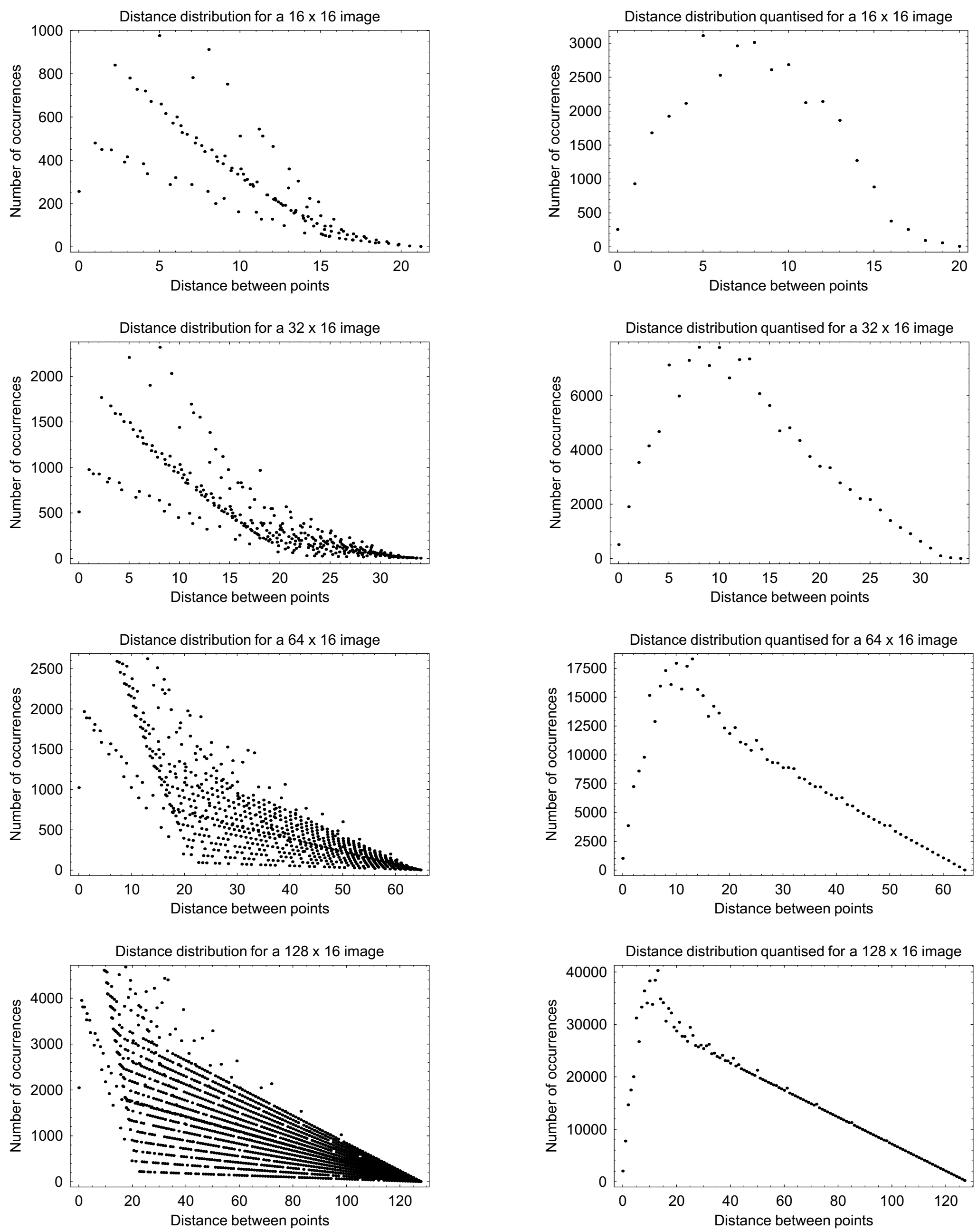


Figure 24: The distribution of actual distances in rectangular windows shown for various aspect ratios.

8.4.2.2 Variation due to changes in aspect ratio

The variability of aspect ratio affects the distribution of distances in a sensor population and therefore has some bearing on JIGSAW.

As the aspect ratio increases from one to infinity (or decreases from one to zero), the curve evolves towards a simple inverse linear distribution (see **Figure 24** and **Figure 25**). The effect of this is that the probability of choosing a mid-length pair decreases and the probability of choosing a short distance pair increases. In isolation, this is probably beneficial to JIGSAW, but the change is not isotropic, since the orientations of longer distances progressively converge towards the major axis of the image, whereas the orientations of shorter distances remain more evenly distributed. We know that high *a* values (actual distances) will probably produce high *b* values, and that high *b* values produce increasingly unreliable *g* values. The concentration of high *a* in one orientation therefore has the significant effect of making, for example, horizontal ordering more difficult than vertical ordering in an image of landscape aspect ratio. This, therefore, exacerbates any horizontal banding effect.

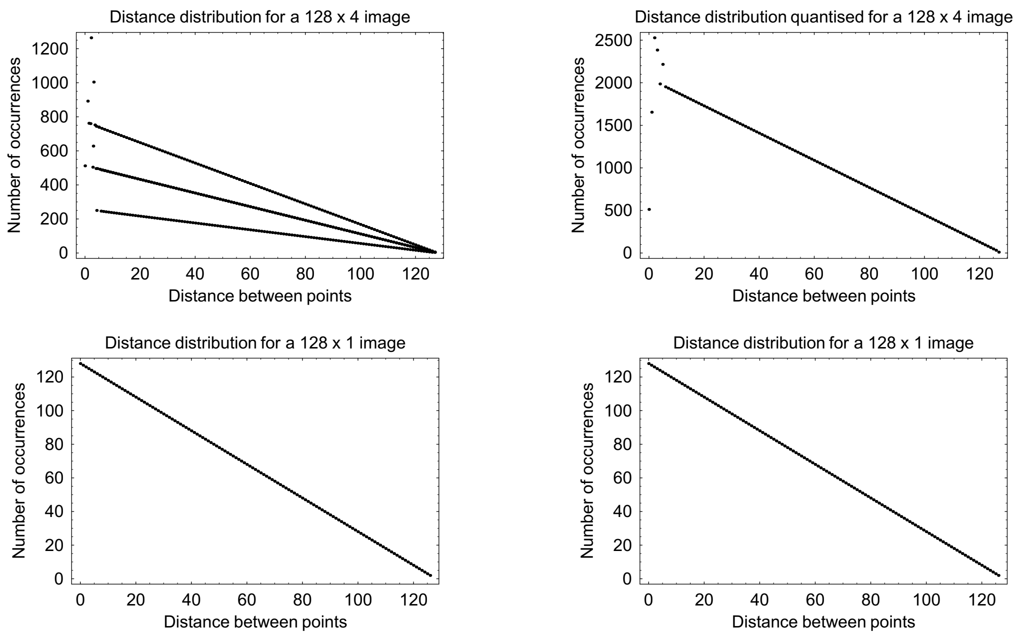


Figure 25: The distribution of actual distances in rectangular windows shown for thin aspect ratios.

In summary, aspect ratio has a bearing on the overall distribution of actual distances; it creates non-uniform angular distributions of lengths.

8.4.2.3 Circular windows

The non-uniformity of the *a* distribution is not a product of corners in the window. We can consider a circular window of sensors where the sensors are evenly distributed. The curve in **Figure 26** estimates the distribution of *a* in a circular window of diameter 1. We plot the distance between randomly selected pairs of points (with a uniform spatial distribution of points). It can be seen from this plot that circular windows are not too dissimilar to square ones.

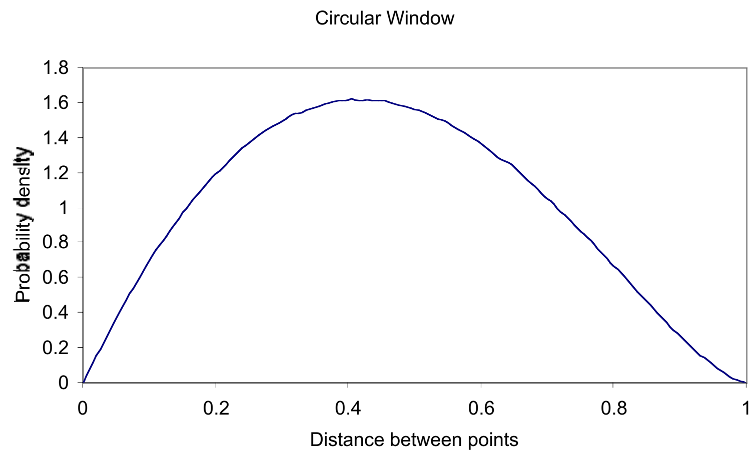


Figure 26: The distribution of actual distances in circular windows.

8.4.2.4 Spherical windows

All of the preceding charts of distributions are taken from bounded windows. An obvious unbounded arrangement of sensors is the full spherical array.

It can be seen in **Figure 27** that even such an ideal arrangement of sensors produces a non-uniform distribution. The chart estimates the distribution of a in a spherical window of radius 1 by plotting the distance between randomly selected, spatially uniform, pairs of points on the surface of a sphere.

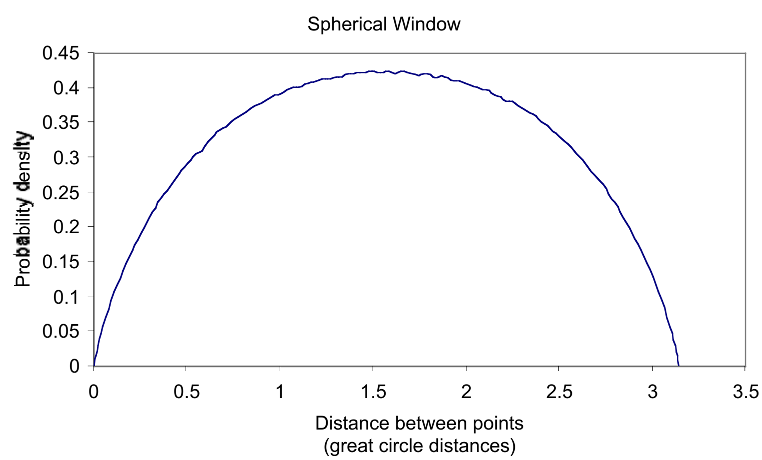


Figure 27: The distribution of actual distances in spherical windows.

8.4.3 Edge effect on distribution of *a*

The longest distance within any rectangle is its diagonal. So only the pixels at the corners of rectangular windows can be paired at this maximum distance. By contrast, the range of distances that can be measured from the centre of a window extends to only half the diagonal. Clearly, the distributions of distances available from fixed points in the window are not the same. The spatial distribution of distances is anisotropic. **Figure 28** is a surface plot of the sum of distances measurable from each point in a 16 × 16 window. Corner points have a value approximately double that of the centre points.

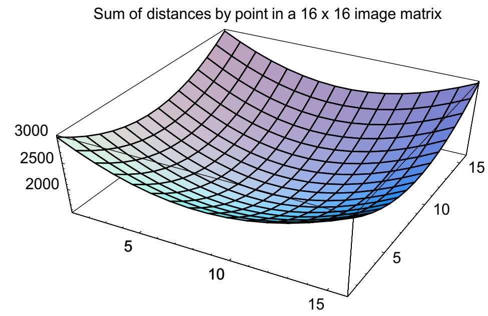


Figure 28: Total actual distances from each point in a 16 × 16 image.

We already know that the *ab* curve is asymptotic. This means that for a high *b* value we are less certain about the actual distance, *a*. This uncertainty increases the magnitude of error, so we observe that corner pixels are generally the last to converge, and never settle into position as well as centrally located ones.

8.5 Noise inherent in the transmission system

In addition to sensor effects that are deliberate, and those that are inescapable, there is noise, which is accidental. While it is common to try to minimise noise, robust algorithms will continue to work with low signal-to-noise ratios.

8.5.1 Video edge effects

Video signals consist of 576 horizontal lines. The top and bottom lines are actually half-empty, or contain half a line of picture, and half a line of spurious signals and noise. In addition to this, the clarity of the picture is usually compromised two or three lines into the image. The left and right edges often show a marked increase or decrease in brightness, and loss of contrast. The effect can be noticeable up to sixteen pixels into the image.

8.5.2 Video noise

Constant noise of about 8 grey levels in a scale of 256 can be seen across the entire image, even when using closed-circuit equipment. Videotape noise can also be present.

8.5.3 Signal interference

Many of the technologies developed for transmitting signals are prone to signal interference. This is where some external signal (natural or artificial) contaminates the signal being transmitted. Transmissions using electromagnetic radiation are particularly prone to signal interference. For example, electrical equipment can generate interfering radiation that gives analogue television signals a certain waviness.

Considering that signal behaviour is the starting point for JIGSAW processing, there is a potential that signal interference will be quite detrimental. However, whether or not signal interference affects JIGSAW is dependent upon the nature of the interference.

In the case of television signals the structure of the signal also encodes pixel location. Interference that affects only this aspect (e.g. waviness) will not affect JIGSAW. For signal interference to affect JIGSAW it must change the signal behaviour in such a way as to distort behavioural differences between pixels (or in general, sensors).

8.6 Summary of sensor system characteristics

In this section, we dealt with the characteristics of sensors systems, with particular attention to visual information transmission systems. The subjects covered were sensor type, sensor modality and three types of transmission effect: artistic deliberation (avoidable), artefacts (unavoidable), and noise (accidental). In dealing with some of these factors we have developed several computational responses, which we will now describe.

9 Computational methods

There are a number of practical issues to consider when implementing JIGSAW. The characteristics of real sensor systems mean JIGSAW may need to cope with artistic deliberation, low signal-to-noise ratios, and complex sensors. There are inescapable artefacts coming from the way sensors encode quantities, and edge effects from the window they form. There are also practical limits on our computing machines.

To deal with the practicalities of the real world, we expand JIGSAW to cover a family of algorithms. Different components can be included, and decisions made, which make tradeoffs between

- the computational power available,
- the complexity of the sensor system and environment,
- assumptions made about sensor system and environment.

There is much more work that can be done to explore enhancements to JIGSAW and its operating characteristics. Below, we detail our initial tactical measures, and some observations concerning implementation issues.

9.1 Geometry dimensionality

A key decision before the JIGSAW implementor is, "How many dimensions should there be in the geometry?" The work we have reported here has concentrated almost exclusively on vision systems, and we have used a two dimensional geometry to correspond to the sensor plane. Such a decision cannot be taken lightly since it could be argued that a third dimension would allow the geometry to bend in compensation for effects of the lens (e.g., wide angle). Assigning a dimensionality to the world, in effect, makes an assumption and needs to be recognised as such.

Adding an extra dimension beyond what is assumed to exist in the environment may provide extra freedom to the gradient descent as it attempts to find a global minimum. We have noticed in some of our experiments a phenomenon where separate *locally* consistent patches of geometry are not themselves *mutually* consistent. For a two dimensional geometry, the result looks rather like a twist (see, for example **Figure 45**). Sometimes the geometry untwists by destroying one consistent patch and rebuilding it in reflection. Other times the twist is a local minimum.

When dealing with abstract spaces, such as concept spaces, there may not be any *a priori* knowable dimensionality. Working with such domains and spaces is an interesting area requiring further research. Also worthy of future research is the possibility of adding or removing dimensions as required.

9.2 Sensor population

All our experiments have so far involved a set of sensors that remains the same throughout processing. An alternative may be to achieve spatial organisation by first using only a subset of the sensors, then to gradually introduce additional sensors. The usefulness of this method requires further investigation.

9.3 Workforce

When a pair of sensors are selected for processing, the various metrics associated with that pair (i.e., b and μ') are brought to bear on their geometry. As will be shown below, there are a number of alternatives for evaluating the metrics and some methods require additional memory and processing for each possible pair of sensors.

We define the metrics associated with a pair as one *unit* in the *workforce* available to the gradient descent. Lightweight units that need no additional memory or processing do not impact on the complexity of the algorithm. For implementations requiring resources to compute metrics, there will be an impact, and in such cases, it makes sense to consider having fewer units than the total number of possible pairs.

9.3.1 Complete workforce

A workforce that has one unit for each pair we call a *complete workforce*. The number of units, w, required for a complete workforce over n sensors is

$$w = \frac{n(n-1)}{2} \tag{22}$$

In this situation, each sensor is associated with n-1 units.

A complete workforce is useful when there are no processing requirements to maintain unit metrics, or when the number of sensors is small. If the processing to maintain each unit is constant, then the complexity of each iteration is $O(n^2)$.

9.3.2 Sampled workforce

The fact that some of the techniques for obtaining b and μ' require additional processing and memory urge us to consider using less than a complete workforce.

When the workforce consists of the entire population of possible pairs, there is much redundancy in the information used to recover the sensor geometry. In principle, a sensor only needs three units to correctly position it in a 2D geometry. While in practice three units per sensor is unlikely to produce good results, it does suggest restricting the workforce to a fixed multiple of n.

A workforce that is a subset of the population we call a *sampled workforce*. If the number of units in the workforce is w, then the average number of units per sensor is $2w/n$ which we refer to as the *order* of the workforce. A complete workforce will be order n-1.

The advantage of a sampled workforce is the reduction in space and time complexity by having a constant order workforce (which is much less than order n-1). A gain is only achieved, though, if there is actual memory and processing required to maintain the units. This may not be the case when instantaneous b values are used (discussed in section 9.5).

To implement a sampled workforce a new question must be answered, "How is a unit of the workforce assigned to a particular pair of sensors?" A simple answer is to assign the workforce to a random sample of the population.

More complicated regimes may exist for assigning a sampled workforce, but only one other is considered here. We consider starting with a randomly sampled workforce which is then evolved by occasionally reassigning units to new sensor pairs.

9.3.3 Evolving a sampled workforce

We can see from the sensitivity analysis of parameters, and the empirical studies of the ab curve, that small values of b are more reliable than higher values of b. An opportunity arises when using a sampled workforce to evolve the workforce to eliminate the unreliable units.

The underlying algorithm for evolving the workforce is,

1) Select a workforce unit, *u*, (for sensors *i* and *j*).
2) Adjust the geometry of *i* and *j* according to the gradient descent formulation.
3) With probability, *p*, reassign *u* to a new pair of sensors.
4) Go to 1.

Different evolution regimes control the probability, *p*, and the method for reassigning *u*.

The basic policy when evolving the workforce is to prefer pairs which have smaller, more reliable values for *b*. This suggests that if *b* for a particular unit is small, then it does not need to be reassigned. Based on this intuition, the following formula for *p* has been tried,

$$p = 1 - \frac{1}{1 + k_{break}\frac{b}{\mu'}} \tag{23}$$

As the *b* metric of a unit increases, so does the probability that the unit will be reallocated. The parameter, k_{break}, controls how likely the high-*b* units are reassigned. We can see in **Figure 29** how this controls the workforce's preference for small *b* values.

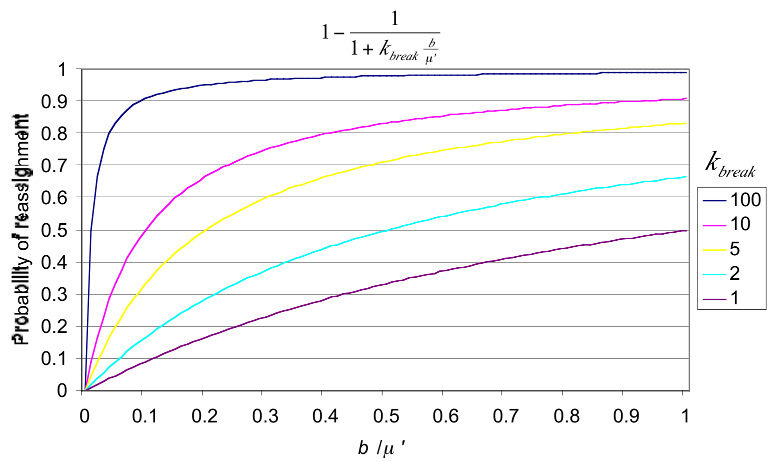


Figure 29: As k_{break} increases, the system more aggressively reassigns units with a high *b*.

Once it has been decided that a unit is available for redeployment, there is still a question of how to reassign it. Simply reassigning it to a randomly chosen pair of sensors is an option which ignores the principle for preferring small *b* values. Ideally, we would like to scan all possible pairs and choose one with low *b*, but this has a number of complications. First, there is the computational expense of such a search. More seriously, there is the problem of evaluating *b*. The reason for using a sampled workforce is to control the computational expense dedicated to evaluating *b*. If redeploying a unit requires *b* to be computed for every possible pair, we lose the advantage.

These problems can be solved by taking a small random sample of possible pairs. Then using the geometrical distance, *g*, as an indicator of *b*, assign the unit to the pair with the lowest *g*. Taking a sample avoids the need to check every possible pair, and using *g* to indicate *b* avoids the time required to accumulate *b*.

Initially the geometry is unlikely to resemble the actual positions of the sensors, so *g* will be a ineffectual indicator of *b*. In such a situation, basing redeployment on *g* is no better than redeploying randomly. However, as the system starts to recover the sensor geometry, there will be an approximately monotonically increasing relationship between *g* and *b*. This occurs because there is a monotonically increasing relationship between a and b, and the system is progressing to equate a and g.

When the redeployment sample size, k_{tries}, is 1, there is no preference for how a unit is reassigned. This is equivalent to just randomly reassigning a unit. As k_{tries} increases, redeployment more aggressively prefers pairs with small g.

In theory, there is a serious, potential problem with aggressively reassigning units to pairs with small g. Units may establish locally consistent patches of geometry which then cannot be pieced together using gradient descent. For example, in trying to join two locally consistent patches, which are in reflected geometries, JIGSAW may encounter a twist that cannot be undone. Alternatively, sensors may get left out of the main cohesive structure, leaving a halo of neglected sensors scattered around the geometry.

The effects of various evolution regimes, and variation in the parameters k_{break} and k_{tries}, are reported in section 10.

9.4 Work agenda

So far we have neglected to discuss how units in the workforce are selected to perform adjustments to the geometry. If the system is implemented with a parallel, distributed processing model, then the question of the *work agenda* is relatively meaningless. However, the ubiquity of serial computers oblige us to address it.

We consider three possible agendas, (1) a linear order, (2) a uniform random order, and (3) a biased random order.

A linear work agenda is straightforward. We simply enumerate the units in the workforce and iterate over them. This has the advantage of ensuring all units are processed equally, but has the disadvantages of requiring working memory to keep track of the next unit, and it imposes an artificial, repetitive ordering.

A variation on the linear order is to randomly select a unit each time, and the most basic selection distribution treats all units uniformly. A uniform, random agenda simulates stochastic parallel processing where only one work unit can adjust the geometry at any one time.

Given that units can be randomly selected from the workforce, it may be useful to bias the selection in some way. One possible bias would be towards units with low b. A biased, random agenda could be useful for complete workforces where the units have no memory allocated (e.g., using instantaneous b values as opposed to accumulated b values).

A biased, random sample is an alternative to evolving the workforce, however it may be difficult to implement. A random agenda may be biased similarly to biasing unit redeployment (i.e., using a parameter similar to k_{tries}).

9.5 Determining *b*

When performing the gradient descent, it is necessary to estimate the actual distance, $a_{i,j}$, for a pair of sensors, i and j. This requires the algorithm to determine $b_{i,j}$ for the pair.

The value $b_{i,j}$ is the average difference in sensor readings for i and j. It is by comparing $b_{i,j}$ to the expected difference, μ, that we can infer $a_{i,j}$.

The cheapest approach to determining $b_{i,j}$ is to take the difference in instantaneous readings of the sensors. The tacit assumption with such an approach is that the overall effect on the geometry will implicitly average $a_{i,j}$ anyway. This assumes the following square commutes:

$$\begin{tikzcd} \widehat{b} \arrow[r, "average"] \arrow[d, "f^{-1}"'] & b \arrow[d, "f^{-1}"] \\ \widehat{a}' \arrow[r, "average"] & a' \end{tikzcd} \tag{24}$$

In this square, $\hat{b}$ represents the measured, instantaneous difference between two sensor values. The vertical arrows represent the function using equation (17).

It is easy to show that the square does not commute. The shape of the *ab* curve means that when an instantaneous $\hat{b}$ varies above *b*, there will be a disproportionately large effect on *a'*, compared to when $\hat{b}$ varies below. However, despite the fact that the square clearly does not commute, surprisingly good results are obtained from using such a simple approach.

A mathematical analysis of the effect of the non-commuting square is yet to be performed. Empirical results are reported in section 10.

To move from instantaneous values to accumulated values requires extra storage and processing time. A simple method for accumulating *b* uses an exponentially decaying average. The average at time *t* can be computed using the formula

$$b_t = \gamma b_{t-1} + \left(1-\gamma\right)\hat{b}_t \qquad (25)$$

where γ is the decay rate and $\hat{b}_t$ is the instantaneous difference at time *t*.

It is interesting, when using this type of average, to ask, "What is a suitable value for b_0?" This is particularly important when the average needs to be initialised part way through the processing (see section 9.3.3).

The solution is to set b_0 to the value that will set the gradient decent factor, *h*, to zero. That is, when the algorithm uses *b* to determine how to change the geometry, there will be no change. As *b* accumulates its estimated value, its effect on the geometry will become more significant.

The change to the geometry will be zero when $g = a'$, i.e.

$$g = -\ln\left(1 - \frac{b}{\mu'}\right) \qquad (26)$$

Therefore, b_0 should be set using

$$b_0 = \mu'\left(1 - e^{-g}\right) \qquad (27)$$

9.6 Determining μ'

The value μ is a statistic that measures the expected behavioural difference between two sensors – independent of distance. The system uses a parameter, μ', to estimate μ. It is by comparing μ' with *b* for a pair or sensors, using formula (17), that it is possible to arrive at an estimate, *a*', of the actual distance.

If the distribution of sensor readings is known for sensors *i* and *j*, then $\mu_{i,j}$ is directly computable. If the probability of reading value *v* is $p_i(v)$ and $p_j(v)$ respectively, then

$$\mu_{i,j} = \sum_x \sum_y p_i\left(x\right)p_j\left(y\right)\left|x - y\right| \qquad (28)$$

In some systems, it may be possible to actually collect the distributions for each sensor. This is a particularly attractive option for sensors which report a small number of discrete values. For each sensor, *i*, a vector, p_i, is required such that $p_{i,v}$ is the probability of sensor *i* measuring value *v*.

Accumulating the frequency data to estimate the probabilities may be difficult due to the temporally open nature of JIGSAW. However, distributions can be accumulated in a time-homogenous way similar to exponentially decaying averages. The following routine decays each probability by a factor of γ. The total decayed amount is added to the probability for the current sensor value. In this way the area under the probability density function is conserved.

The initial distribution could be uniform, in which case $p_{i,v}$ is initialised to $1/r$, where r is the number of possible sensor values. With decay rate γ, the update routine for p_i is:

1) For each sensor, *i*,
   1.1) Set the decay total = 0
   1.2) For each value, *v*,
      1.2.1) Set the decay = $\gamma\, p_{i,v}$
      1.2.2) Set $p_{i,v} = p_{i,v}$ – decay
      1.2.3) Set total = total + decay
   1.3) Let *v* be the current value of the sensor, *i*
   1.4) Set $p_{i,v} = p_{i,v}$ + total

If it is known that the sensors readings are uniformly distributed between 0 and r-1, then calculating μ simplifies to

$$\begin{aligned}\mu &= \frac{1}{r^2}\sum_{x=0}^{r-1}\sum_{y=0}^{r-1}|x-y| \\ &= \frac{r^2-1}{3r} \\ &\approx \frac{r}{3}\end{aligned} \tag{29}$$

The approximation holds for large values of r, and will not underestimate. In general, the expected difference for a pair of uniform discrete distributions (from α to β) is given by

$$\begin{aligned}\mu_{i,j} &= \sum_{x=\alpha_i}^{\beta_i}\sum_{y=\alpha_j}^{\beta_j}\frac{1}{\left(1+\beta_i-\alpha_i\right)}\frac{1}{\left(1+\beta_j-\alpha_j\right)}|x-y| \\ &= \frac{\left(\begin{array}{l}|\alpha_i-\alpha_j|-|\alpha_i-\alpha_j|^3+|\beta_i-\beta_j|-|\beta_i-\beta_j|^3 \\ +\ \alpha_i+\alpha_j-\beta_i-\beta_j+(1-\alpha_i+\beta_j)^3+(1-\alpha_j+\beta_i)^3-2\end{array}\right)}{6\left(\alpha_i-\beta_i-1\right)\left(\alpha_j-\beta_j-1\right)}\end{aligned} \tag{30}$$

The derivation of this result is explained in more detail in appendix 2.

For sensors which report real-number values, the equivalent to formula (28) requires $p_i(v)$ and $p_j(v)$ to be the probability density functions for sensors. The formulation is then

$$\mu_{i,j} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p_i(x)\,p_j(y)\,|x-y|\; dx\, dy \tag{31}$$

For practical purposes, this is not particularly useful. However, distributions for both discrete and continuous sensors can be approximated by categorising readings into groups of ranges, then using formula (28) over the mid-points of each range. In the most basic case, only two ranges exist, one from the minimum value to some threshold, and the other from the threshold to the maximum. This bears some similarity to the representation of pictures as binary bit maps. Of course, the quality of such a simplification is highly dependent upon the choice of threshold value.

When real-number sensor reading are uniformly distributed between α and β, then μ has a direct solution,

$$\mu_{i,j} = \int_{\alpha_i}^{\beta_i}\int_{\alpha_j}^{\beta_j} \frac{1}{(\beta_i - \alpha_i)} \frac{1}{(\beta_j - \alpha_j)} |x - y| \, dx \, dy$$
$$= \frac{3\left(\alpha_i\alpha_j + \beta_i\beta_j + \alpha_i\beta_j + \alpha_j\beta_i - \alpha_j^2 - \beta_j^2\right) - 2\left(\alpha_i^2 + \beta_i^2 + \alpha_i\beta_i\right)}{6\left(\alpha_j - \beta_j\right)} \tag{32}$$

The derivation of this result is explained in more detail in appendix 3.

Formula (31) also becomes tractable when it is known that the sensor readings are normally distributed. Given the mean, v_i, and standard deviation, s_i, for each sensor, i, the probability of obtaining various readings at sensor i is given by the normal distribution probability density function,

$$p_i(x) = \frac{e^{-\frac{(x - v_i)^2}{2 s_i^2}}}{s_i\sqrt{2\pi}} \tag{33}$$

By considering the z-scores of integration variables x and y, formula (31) becomes

$$\mu_{i,j} = \frac{1}{2\pi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-\frac{1}{2}\left(x^2 - y^2\right)} \left|\left(v_i + s_i x\right) - \left(v_j + s_j y\right)\right| \, dx \, dy \tag{34}$$

Symmetries in the problem allow this to be further simplified by considering the expected difference of one theoretical sensor from a zero point. This theoretical sensor would have mean value, $v_{i,j}$, and standard deviation, $s_{i,j}$, given by

$$v_{i,j} = \left|v_i - v_j\right| \tag{35}$$

$$s_{i,j} = \sqrt{s_i^2 + s_j^2} \tag{36}$$

Then we have

$$\mu_{i,j} = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} e^{-\frac{1}{2}x^2} \left|v_{i,j} + s_{i,j}x\right| \, dx$$
$$= \sqrt{\tfrac{2}{\pi}}\, e^{-\frac{1}{2}\left(\frac{v_{i,j}}{s_{i,j}}\right)^2} s_{i,j} + \mathrm{Erf}\left(\frac{1}{\sqrt{2}}\frac{v_{i,j}}{s_{i,j}}\right) v_{i,j} \tag{37}$$

where $\mathrm{Erf}(x)$ is defined as the integral of the Gaussian distribution,

$$\mathrm{Erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2} \, dt \tag{38}$$

A more detailed derivation of this result is covered in appendix 4.

We do not know of any simple solution to the Erf integral. A practical JIGSAW implementation may use a table of externally computed Erf values to approximate it. Alternatively, we note that it can be approximated by

$$\mathrm{Erf}(x) \approx 1 - e^{-\sqrt{2}x} \tag{39}$$

There are more complicated methods for producing an accurate approximation to this integral, however, using this approximation, we have

$$\mu_{i,j} \approx \sqrt{\frac{2}{\pi}} e^{-\frac{1}{2}\left(\frac{v_{i,j}}{s_{i,j}}\right)^2} s_{i,j} + \left(1 - e^{-\frac{v_{i,j}}{s_{i,j}}}\right) v_{i,j} \tag{40}$$

This approximation will not overestimate $\mu_{i,j}$ by more than 0.025 $v_{i,j}$, nor underestimate it by more than 0.095 $v_{i,j}$. A demonstration of this approximation is provided in appendix 5.

The JIGSAW algorithm is sensitive to the parameter μ', and the various formulas above may be used to provide an estimate. Each makes assumptions about the sensor readings and requires a degree of processing power and memory.

We note that if only the sensors' range is known, a safe approach is to set μ' to the maximum possible difference between sensor readings. In all likelihood, this will always overestimate. However, in the absence of any other information this may be better than risking an underestimated μ'.

We are left with a list of possible methods for obtaining μ'. Which one is chosen will depend on how much *a priori* knowledge the system designer has of the environment, how much computational power is available, and how good the result needs to be. The requirements, advantages and disadvantages are summarised in **Table 1**.

Method	**Requirements and repercussions**
Domain Expert Given appropriate domain knowledge, set μ' to μ.	• assumes μ is spatially uniform • there is a risk that μ is underestimated • such knowledge may not be available
Cheap and Safe Given only the sensor range, set μ' globally to be the maximum possible *b*.	• assumes μ is spatially uniform • μ cannot be underestimated • μ will be overestimated
Economy Assume a particular distribution (uniform or normal). Compute and store the distribution parameters for each sensor (e.g., mean and standard deviation). Use these to dynamically estimate μ.	• does not assume μ is spatially uniform • sensor values may not have the assumed distribution • extra processing power is required for each sensor
Approximate Distributions Divide the sensor range up into sub-ranges. Collect the distributions for the sub-ranges and use them to dynamically estimate μ.	• does not assume μ is spatially uniform • need to determine the ranges *a priori* • extra processing power is required for each sensor
Premium Collect the actual value distribution for each sensor. Use them to dynamically estimate μ.	• does not assume μ is spatially uniform • does not assume a particular distribution • processing requirements may be prohibitive

Table 1: Possible methods for obtaining μ'.

9.7 Soft and hard *b* limits

Considering the evidence that large *b* values are less reliable than small ones, an obvious extension is to reduce the effectiveness of units with large *b*. We saw above that one method for preferring small b values is to evolve the workforce. For complete workforces this is not possible. An alternative is to limit the effectiveness of a work unit according to b.

Any time the value of b goes above μ', formula (17) cannot be computed, so such a unit becomes ineffective. This can be extended by providing a hard cut-off value, k_{cut}, such that whenever b/μ' is greater than k_{cut}, no adjustment to the geometry is made by that unit. Implicitly the default value for k_{cut} is 1. Any value less than 1 will reduce the effective workforce.

Rather than having a hard cut-off for b values, a softness factor can be included in the gradient descent such that as b increases, the force with which sensors are geometrically repositioned is reduced. A natural candidate softness formula is

$$\frac{1}{1 + k_{soft}\frac{b}{\mu'}} \tag{41}$$

This is deliberately similar to formula (23) for evolving the workforce. Softness can be used in the gradient descent algorithm in the following way:

1) Select a pair of sensors, i and j
2) Ascertain b, μ' and g for the pair
3) Use b and μ' to derive a'
4) Compute *factor* = k (1 - a' / g) (1 / (1 + k_{soft} b / μ'))
5) For each dimension, x, in the geometry,
   5.1) Compute $h = (x_j - x_i)$ *factor*
   5.2) Set x_i to $x_i + h$
   5.3) Set x_j to $x_j - h$
6) Go to 1.

We currently have no systematic method for selecting values for k_{cut} or k_{soft}.

9.8 Other hill climbing formulations

We note that gradient descent is only one form of hill climbing and that other greedy optimisation algorithms may work to register g with a. The heart of the gradient descent, as we have formulated it, is the factor of h (see equation (10)), which is

$$1 - \frac{a'}{g} \tag{42}$$

When $g > a'$ this factor is positive, which works to move the geometrical representations of the sensors closer together. Conversely, when $g < a'$ this factor is negative, which works to move them further apart. This can easily be conceptualised as a force that operates on the sensors' geometrical representation.

This formula has a certain asymmetry about it. For a given difference between a' and g, the force to push sensors apart is stronger than the force that would attract them.

Our experiences with some versions of JIGSAW showed a halo effect, where sensor representations were pushed outside the main geometry, and became stuck in a halo. This observation prompted us to consider other formulations in the same spirit of the gradient descent. However, all have been limited in overall effectiveness for JIGSAW. We summarise them in **Table 2**.

$\ln\left(\frac{g}{a'}\right)$	This formulation creates symmetry between the repulsive and attractive forces of a unit. Taking the log means that the force is less than proportional to difference between *a*' and *g*.
$g - a'$	This formulation creates symmetry between the repulsive and attractive forces of a unit. The force is proportional to difference between *a*' and *g*.
$\frac{g}{a'} - 1$	This formulation is in some sense, opposite to the original. For a given difference between *a*' and *g*, the force to pushing sensors apart is weaker than the force which would attract them.

Table 2: Some alternative hill-climbing factors.

9.9 Summary of computational methods

There are a number of questions to be asked by a JIGSAW implementor. These include,

- What is the dimensionality of the geometry?
- What is the population of sensors to be processed and how will this change?
- What method will be used for determining *b*?
- What method will be used for determining μ'?
- Will hard or soft limits be used to ameliorate errors in *b* or μ', and if so, what should the parameters be set to?
- Will the workforce be the complete set of sensor pairs or a sampled workforce?
- If the workforce is sampled, will the workforce be evolved to ameliorate errors in *b* or μ', and if so, what should the evolution parameters be set to?

Two obstacles which face a JIGSAW implementor are (1) the need to approximate μ, and (2) the *ab* relationship may be less than ideal due to non-linearities in the sensors or causal relationships. In any case, evidence shows that the smaller *b* is, the more reliable is the relationship. From this we offer the heuristics for limiting *b* and evolving the workforce. To approximate μ we have offered the methods in section 9.6, however these require additional processing and have parameters to set.

Limits to the available computational resources present a third obstacle. In an austere implementation of JIGSAW, *b* values are not accumulated and μ' is not dynamically calculated. In this case, there is no processing power or memory allocated to workforce units (apart from adjusting the geometry) and the workforce is complete. As a consequence, the only means available for ameliorating errors are *b* limits.

10 Simulations and tests

JIGSAW is designed to continuously organise the geometry for interpreting sensor signals based on signal behaviour. In this technical report we have divided JIGSAW into two major components:

1. producing an estimate, *a*', of the distance between sensors based on behavioural difference, *b*,
2. adjusting the geometry based on *a*'.

The first component can be further divided into:

1.1. collecting behavioural difference, *b*,

1.2. determining the parameter, μ',

1.3. using a function to determine a', where $a' = f^{-1}(b)$.

Some of the simulations used to test JIGSAW aim to isolate particular components. Other simulations informally show the sensitivity to implementation decisions.

All the simulations presented here are in the domain of computer vision. We take some source image (either a static image or dynamic video) and present it to JIGSAW with an initially random geometry. The static image simulations use synthesised b values.

To provide a metric for how well a JIGSAW simulation has performed we need some measure of the difference between the actual positions and geometrical representation of sensors. Unfortunately, the error measurement that was used for the gradient descent formulation is inadequate, as it is scale-dependent. Instead we introduce a new term, incoherence, δ, which measures the variance in scaling between a and g for all sensors. Incoherence is given by the formula

$$\delta = \mathrm{var}_{i,j}\left(\ln\left(\frac{a_{i,j}}{g_{i,j}}\right)\right) \tag{43}$$

where $\mathrm{var}_{i,j}$ is the variance over the population of sensor pairs, i and j.

When δ is zero, the geometrical representation is fully coherent with the actual positions, with the possible exception of translation, rotation, reflection, and scale. As the disparity between a and g increases, δ increases.

The meanings of all parameters reported in this section are summarised in **Table 3**.

image size	The number of pixels in the source image (*width* × *height*)
sequence	Name of the source images
iterations	The number of units (sensor pairs) selected for processing. It is a measure of the amount of work performed.
incoherence	The variance in scaling between a and g over pairs of sensors.
workforce	Describes the number of units in the workforce. This will either be *complete* or for a sampled workforce will give the *order*.
evolution	Describes how a sampled workforce is evolved in terms of k_{break} and k_{tries}.
agenda	How units are selected from the workforce.
iterations per frame	The number of units selected for processing before changing to the next frame in the image sequence.
descent rate	This is parameter k in the gradient descent.
estimated μ	The method for producing the estimate, μ', for μ. All the simulations presented here use a global, constant value.
estimated a	The formula used to derive a' from b.
b limits	Describes limits on the effects of units with high b. It is characterised in terms of k_{soft} and k_{hard}.
algorithm	The hill-climbing factor which is used.

Table 3: Summary of parameters in simulations and tests.

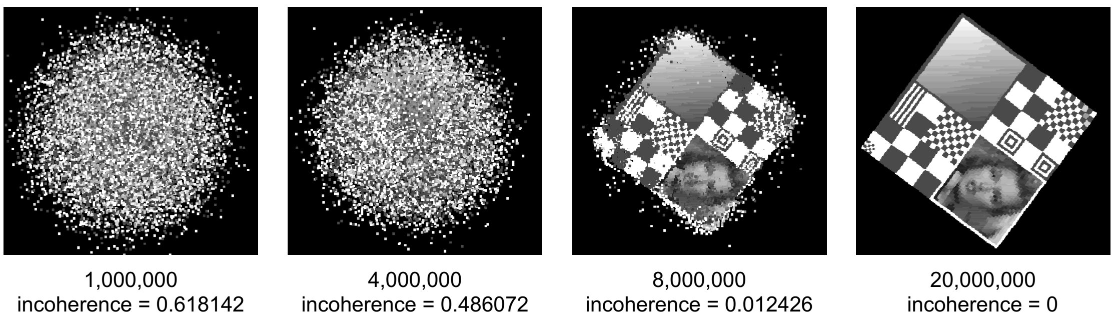
10.1 Static scenes

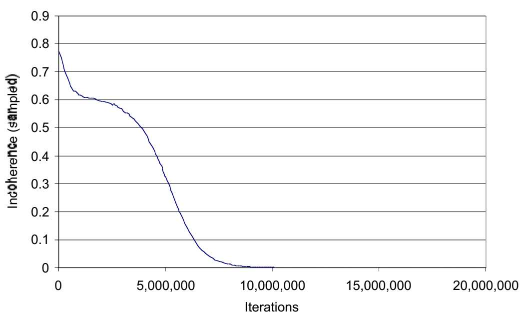
In this section, we report on simulations that present a static image to JIGSAW. As static images do not provide the opportunity to collect behavioural differences, the value for *b* is synthesised for each pair of pixels. The synthesised *b* is produced using formula (16). The purpose of these simulations is to test JIGSAW using controlled *b* values.

The input image serves only to define the set of sensors and provides a visual indication of the algorithm's performance. In all the simulations presented here, the input picture is "test128" shown in **Figure 41**.

10.1.1 Experiment S0 – perfect knowledge of the *ab* relationship

In the simplest simulation, we synthesise *b* values using formula (16) and let JIGSAW have perfect knowledge to derive *a*. In other words, JIGSAW uses formula (17) with $\mu' = \mu$. This test demonstrates that the gradient descent algorithm can work given ideal circumstances. The results show the incoherence quickly falls to zero (see **Figure 30**).



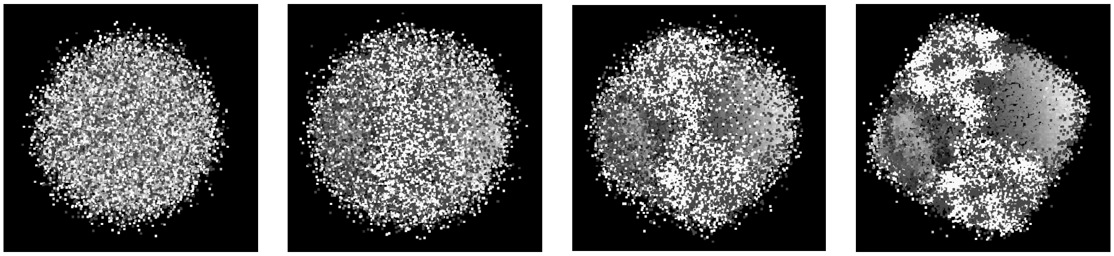


Program Parameters	
image size	128 × 128
workforce	order 100
agenda	random
descent rate	0.05
estimated μ	μ
estimated *a*	-ln (1 – b/μ')
algorithm	1 – a'/g
evolution	none
b limits	none

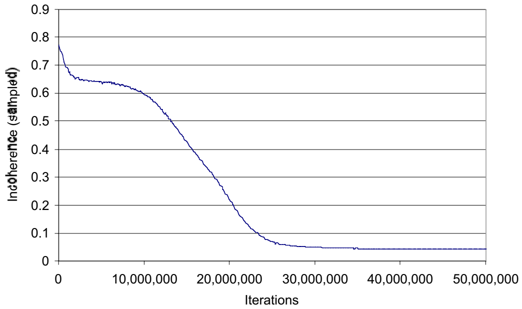
Figure 30: In this simulation, JIGSAW has perfect knowledge of the *ab* relationship.

10.1.2 Experiment S1 – overestimated μ

In the next simulation, we provide JIGSAW with an overestimated μ. The results in **Figure 31** show μ overestimated by a factor of 2. It can be seen that the incoherence measure falls to some low value after which no improvement is seen. As μ is overestimated, *a* will be underestimated the larger *a* is. Consequently, the workforce units try to pull the edges in further than they need to be and the whole geometry becomes distorted and slightly unstable.



1,000,000	15,000,000	20,000,000	50,000,000
incoherence = 0.689965	incoherence = 0.424164	incoherence = 0.220313	incoherence = 0.044849



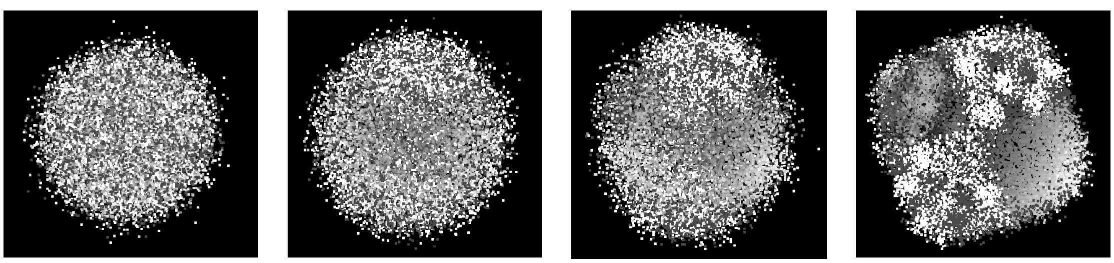
<u>Program Parameters</u>

image size	128 × 128
workforce	order 100
agenda	random
descent rate	0.05
estimated μ	2 μ
estimated *a*	-ln (1 – b/μ')
algorithm	1 – a'/g
evolution	none
b limits	none

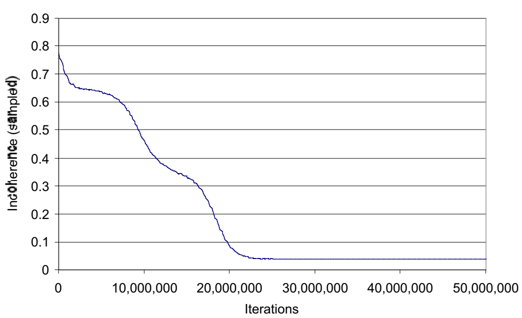
Figure 31: Here we see residual incoherence resulting from JIGSAW having incomplete knowledge about the *ab* curve - the parameter μ' has been overstated by a factor of 2.

10.1.3 Experiment S2 – evolving units to small *b* values

It was shown in sections 6 and 7 that the *ab* relationship is more reliable at small *b* values, even when μ is incorrectly set. Section 9 presented some methods for applying this heuristic. **Figure 32** shows the results of using k_{break} to evolve the workforce towards units with small *b* values.



1,000,000	10,000,000	15,000,000	50,000,000
incoherence = 0.687841	incoherence = 0.461388	incoherence = 0.330101	incoherence = 0.038856



<u>Program Parameters</u>

image size	128 × 128
workforce	order 100
agenda	random
descent rate	0.05
estimated μ	2 μ
estimated *a*	-ln (1 – b/μ')
algorithm	1 – a'/g
evolution	k_{break} = 0.1, k_{tries} = 1
b limits	none

Figure 32: This shows a minor improvement to the residual incoherence created from overstating μ'. The improvement has come from evolving the workforce using k_{break}.

It can be seen that there is a small improvement in the final incoherence value as well as a minor subjective visual improvement. The problem is that using k_{break} alone to evolve the workforce is insufficient to prefer small b values. The parameter k_{tries} is set to 1 which means units are redeployed with no preference. **Figure 33** shows the distribution of b across units and how this changes with the given evolution policy.

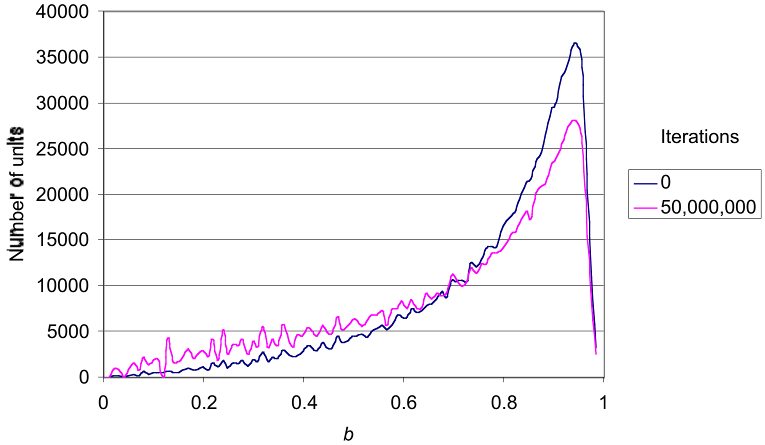
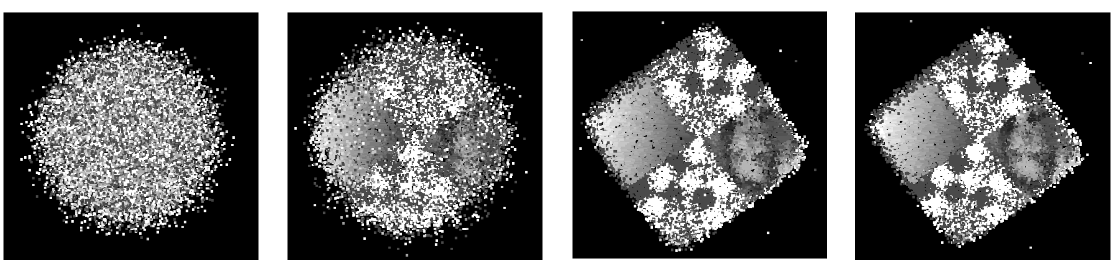


Figure 33: Evolution of the distribution of b across units in the workforce. Evolution parameters are k_{break} = 0.1, k_{tries} = 1.

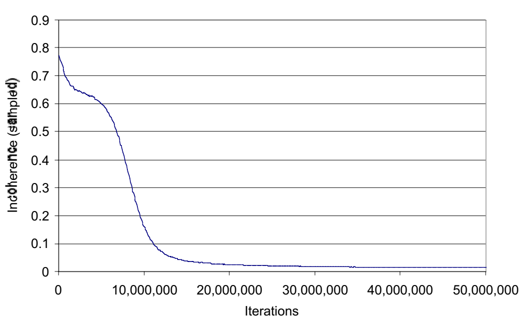
To improve the evolution towards smaller b values, the simulations reported next use $k_{tries} > 1$.

10.1.4 Experiment S3 – redeploying units preferring small *g*

In this simulation, k_{tries} is set to 2 which introduces a slight preference for shorter g values when redeploying units. (See section 9.3.3 for a discussion on using g as a surrogate for b when evolving the workforce.) From **Figure 34** we can see that incoherence drops more quickly to an approximate solution, then there is a slower reduction in incoherence as the workforce evolves to smaller b values. **Figure 35** shows how the population has evolved.



1,000,000	10,000,000	50,000,000	100,000,000
incoherence = 0.685115	incoherence = 0.15948	incoherence = 0.014703	incoherence = 0.01371



Program Parameters	
image size	128 × 128
workforce	order 100
agenda	random
descent rate	0.05
estimated μ	2 μ
estimated *a*	-ln (1 – b/μ')
algorithm	1 – a'/g
evolution	k_{break} = 0.1, k_{tries} = 2
b limits	none

Figure 34: This simulation show a significant improvement in performance by using k_{tries} to redeploy units with a slight preference for small g values.

In the last frame of **Figure 34**, we see some outlying pixels that are unusually distant from their correct positions. This leaves a residual incoherence, which is unlikely to be reduced further. The pixels become isolated because units are aggressively redeployed.

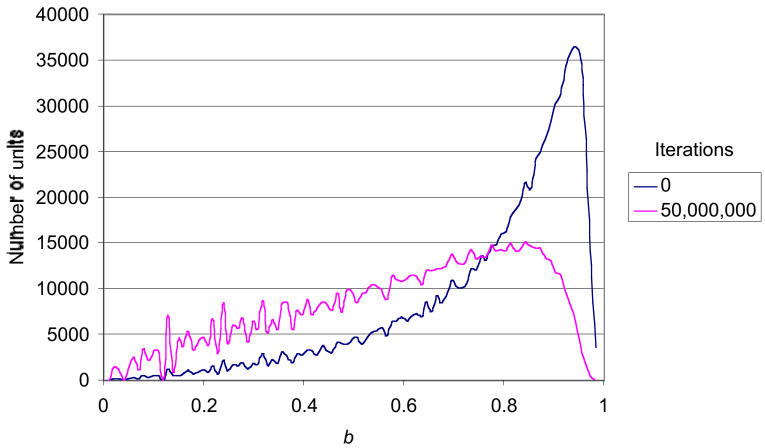
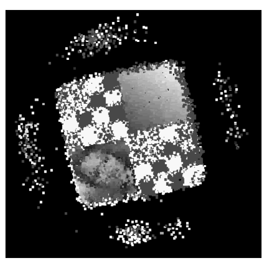
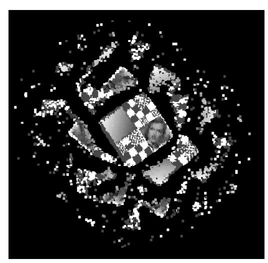


Figure 35: Evolution of the distribution of *b* across units in the workforce. Evolution parameters are k_{break} = 0.1, k_{tries} = 2.

To get the benefits of units with small *b*, there is a need to reduce the unit-breaking rate to allow a partial global solution to be found relatively early. The severe effects of fast evolution (using *g* values) can be seen in **Figure 36**. The first image shows a large number of isolated pixels forming a halo about the central coherent picture. The second image shows how aggressive evolution causes multiple, locally coherent patches to form.



200,000,000
incoherence = 0.06554
evolution: k_{break} = 10, k_{tries} = 2



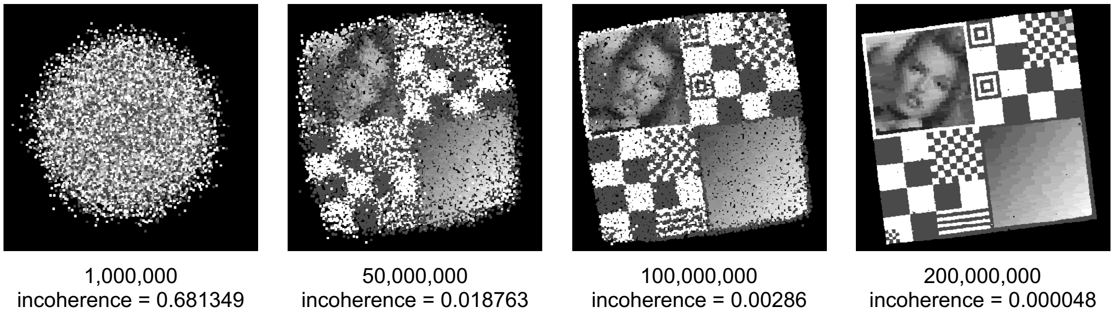
200,000,000
incoherence = 0.40541
evolution: k_{break} = 10, k_{tries} = 10

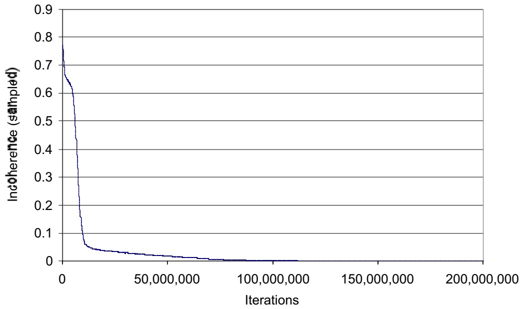
Figure 36: This shows the effect of evolving the workforce too quickly to prefer short *g* values. All program parameters are the same as those given in Figure 34, with the exception of the workforce evolution policy.

10.1.5 Experiment S4 – successfully, aggressively preferring small *g*

To ameliorate the effects of aggressive evolution, without losing the benefits of evolving to small *b* values, we choose a low breaking rate yet maintain a high preference for small *g* values.

In **Figure 37**, we see rapid progress to a partial solution simply using the initially random workforce. After this, the effect of workforce evolution progresses to reduce incoherence to a negligible level.





Program Parameters	
image size	128 × 128
workforce	order 100
agenda	random
descent rate	0.05
estimated μ	2 μ
estimated *a*	-ln (1 – b/μ')
algorithm	1 – a'/g
evolution	k_{break} = 0.05, k_{tries} = 50
b limits	none

Figure 37: Once again μ' has been overstated by a factor of 2. Despite inaccurate knowledge of the *ab* relationship, evolving the workforce has produced near perfect results.

It can be seen in **Figure 38** that although evolution is slow, the resulting population is heavily biased towards small *b* values. This explains the good results obtained in the simulation.

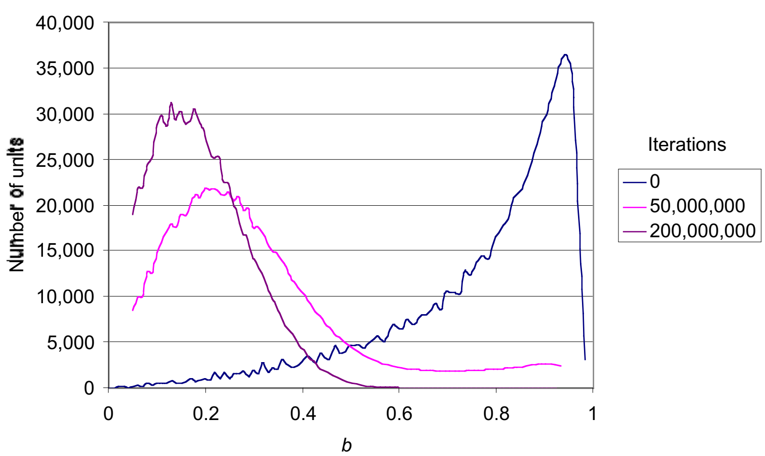


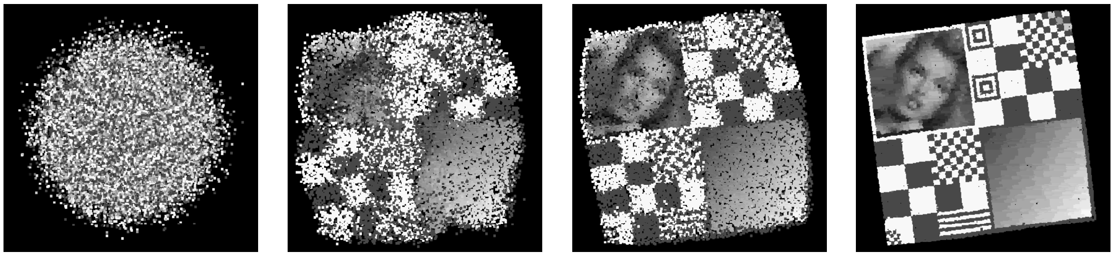
Figure 38: Evolution of the distribution of *b* across units in the workforce. Evolution parameters are k_{break} = 0.05, k_{tries} = 50.

Evolving the workforce in this way has some similarities with simulated annealing (Kirkparick et. al., 1983). As the process progresses, the smaller *b* values lead to smaller *a* values, and consequently the forces changing the pixel geometry become smaller. Workforce evolution has some advantages over simulated annealing. We note that workforce evolution operates time-homogenously and without a global temperature variable. This means that the system can automatically compensate if the geometry needs to adjust to some future perturbation.

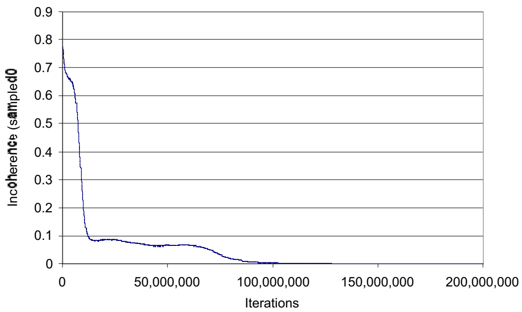
10.1.6 Experiment S5 – ignoring the *ab* relationship

Preferring a workforce with small *b* values is so effective at desensitising the algorithm to distortions in the *ab* relationship, that simply setting *a*' to *b* is sufficient to produce good results.

In the final static-scene simulation presented in **Figure 39**, the results are as good as those obtained using formula (17) to estimate *a*'.



1,000,000	70,000,000	100,000,000	200,000,000
incoherence = 0.703579	incoherence = 0.049858	incoherence = 0.004274	incoherence = 0.00015



<u>Program Parameters</u>	
image size	128 × 128
workforce	order 100
agenda	random
descent rate	0.05
estimated *a*	*b*
algorithm	$1 - a'/g$
evolution	$k_{break} = 0.05$, $k_{tries} = 50$
b limits	none

Figure 39: This simulation shows the effectiveness of workforce evolution to desensitise JIGSAW to inaccurate knowledge about the *ab* relationship. The value *b* is used directly for the estimate, *a*', yet incoherence is reduced to a negligible level.

10.1.7 Conclusions from tests using static scene

Performing simulations using synthesised *b* values has allowed us to control the quality of the *ab* relationship and explore the effect of certain parameters and implementation decisions. There are a number of points we can draw from the simulations:

- With perfect knowledge of the *ab* relationship, JIGSAW can produce perfect results.
- Evolving the workforce to prefer small *b* values can be used to compensate for inaccurate knowledge of the *ab* relationship.
- If units are redeployed too rapidly, using *g* as a surrogate for *b*, then individual sensors, or locally coherent sets, can be left isolated from the main coherent set.

The next section reports on our experiences where JIGSAW was required to obtain its own *b* values.

10.2 Dynamic scenes

JIGSAW has resulted from our belief that signal behaviour is sufficient to organise sensor representation geometry. It is because of this that JIGSAW has been designed to take dynamic input. Problems we noted in static scenes, such as haloes and twists, are considerably less common in dynamic scenes. Our hypothesis is that the highly variable *b* values and (to date) randomised work unit selection help to circumvent situations in which a set of units can lock up a pixel in an outlying position (halo effect) or tie a set of pixels in a knot (twists) from which there is, for a long time, no exit.

10.2.1 Accumulated and instantaneous behaviour

If there is some capacity to accumulate sensor values, then we can compute exponentially decaying averages, derivatives or other values with which to make comparisons, instead of instantaneous b. By accumulating sensor value distributions, μ can be calculated for each work unit. What follows, however, is a discussion of experiments that used instantaneous values of b exclusively.

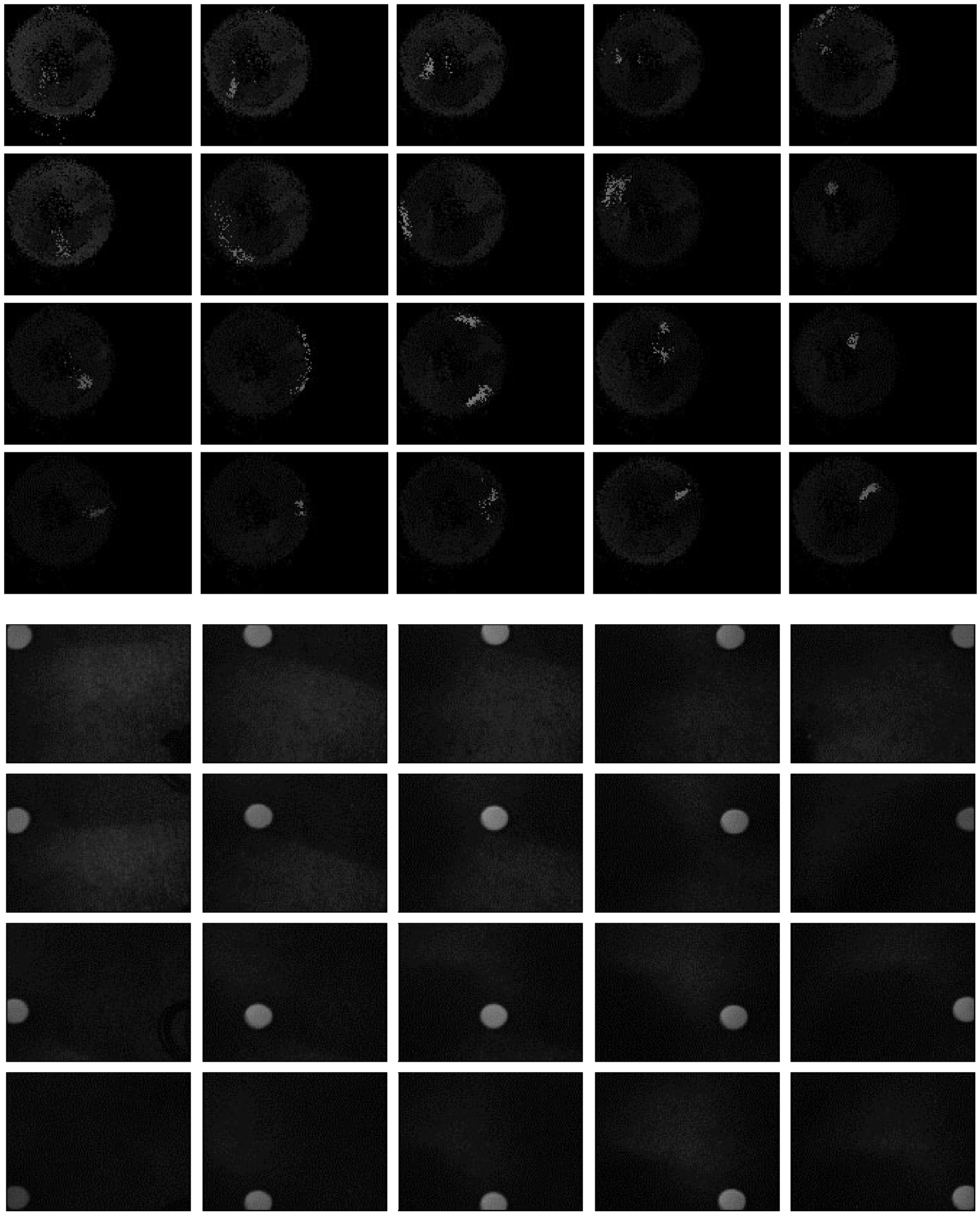


Figure 40: Spatial distortions due to talking head effect in television signals.

10.2.2 Plain TV

Initial experiments with plain television sequences revealed a curious effect, in which the geometry became rounded, and all edges concentrated in the centre as shown in **Figure 40**. In this figure, there are two sets of images. The upper four rows show the geometrical output corresponding to the lower four rows, which are actual scenes (depicting a white disc on a dark background). Each actual scene was passed through a geometry derived from a sequence of broadcast television (Australian Rules Football). The input pictures have been framed so that the disc is visible in 20 different positions in the window.

A number of facts can be determined by looking at these images. Firstly, the geometry is circular, not rectangular like the input. Secondly, the geometry is everted so that when the white disc is at the edge in an input image it shows up near the middle of the geometry of the output image, and *vice versa*. Thirdly, there is some degree of bilateral symmetry in the geometry. The first two effects we believe to be related, since the eversion will produce a circular geometry anyway. The third effect remains unexplained.

The eversion was thought to be caused by the talking head effect, the high similarity of edges in human-framed images (discussed in section 8.3.1). This led us to perform a sequence of controlled experiments in which we attempted to isolate and specify the responsible phenomenon.

In the following series of experiments (D0 to D5), we allowed JIGSAW to run for 500 million iterations. One iteration is a single pair comparison. However, many of these comparisons were discarded for computational reasons, and so had no effect on the geometry. We took snapshots of the geometry evolution at regular intervals, and recorded the incoherence value at the same time. Though each experiment used a different sequence of images, we used a standard test image, "test128", to display the development of all geometries. All the experiments in this series used the same initial data set. There is one exception on both counts. Because of the 4:3 aspect ratio of its source sequence, experiment D0 had a smaller initial geometry and a different test image ("test12896"). The initial geometries had very similar incoherence measures: 0.782576 and 0.790826, respectively. The test images and initial geometries are shown in **Figure 41**.

In each of the presentations of experimental data that follow, we give four randomly chosen examples of input data, four stages in the evolution of geometry, each with an incoherence metric and an iteration count, a graph of μ versus edgeness (explained in section 8.3.1), and the key experimental parameters.

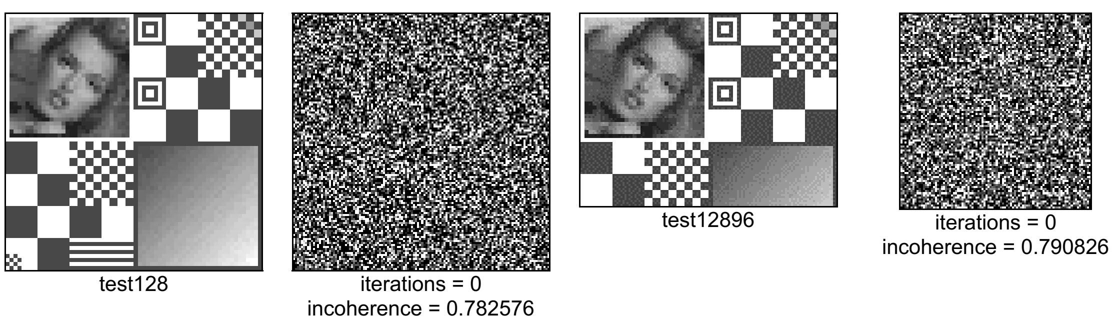


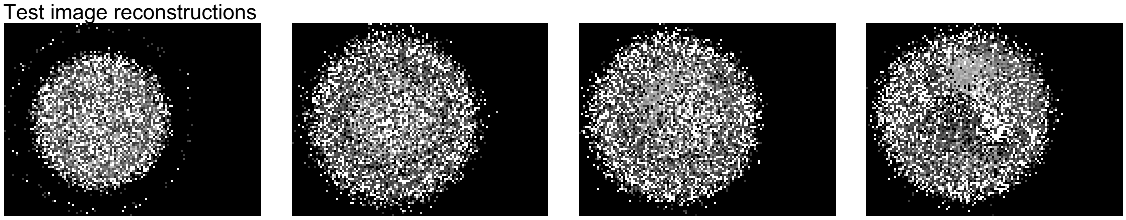
Figure 41: Standard test images and how they look when passed through initial geometries used in the following sequence of experiments.

10.2.3 Experiment D0 – Chomsky video

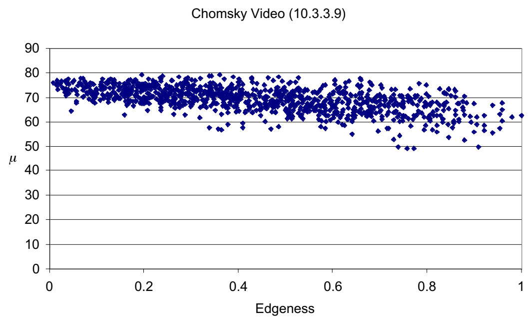
We first replaced the preliminary television test (see **Figure 40**) with a more controlled base. In this, we used the video *Manufacturing Consent: Noam Chomsky and the Media* (Ackbar and Wintonick, 1992) for the input sequence. Video is a cleaner source of images than television, given our poor local reception, so this eliminated much unwanted noise. Though later experiments were conducted using as aspect ratio of 1:1, these images have a 4:3 aspect ratio. This matched the proportions of the original video, and therefore allowed us to use all the edge pixels that we believe are crucial to demonstrating edge effects such as the talking head effect.

Input image samples

Test image reconstructions



125,000,000	250,000,000	375,000,000	500,000,000
incoherence = 0.735762	incoherence = 0.744097	incoherence = 0.697371	incoherence = 0.525124



Program Parameters	
image size	128 × 96
sequence	Chomsky video
workforce	complete
agenda	random
iterations per frame	500
descent rate	0.001
estimated μ	255
estimated *a*	$-\ln(1 - b/\mu')$
algorithm	$1 - a'/g$

Figure 42: Chomsky video.

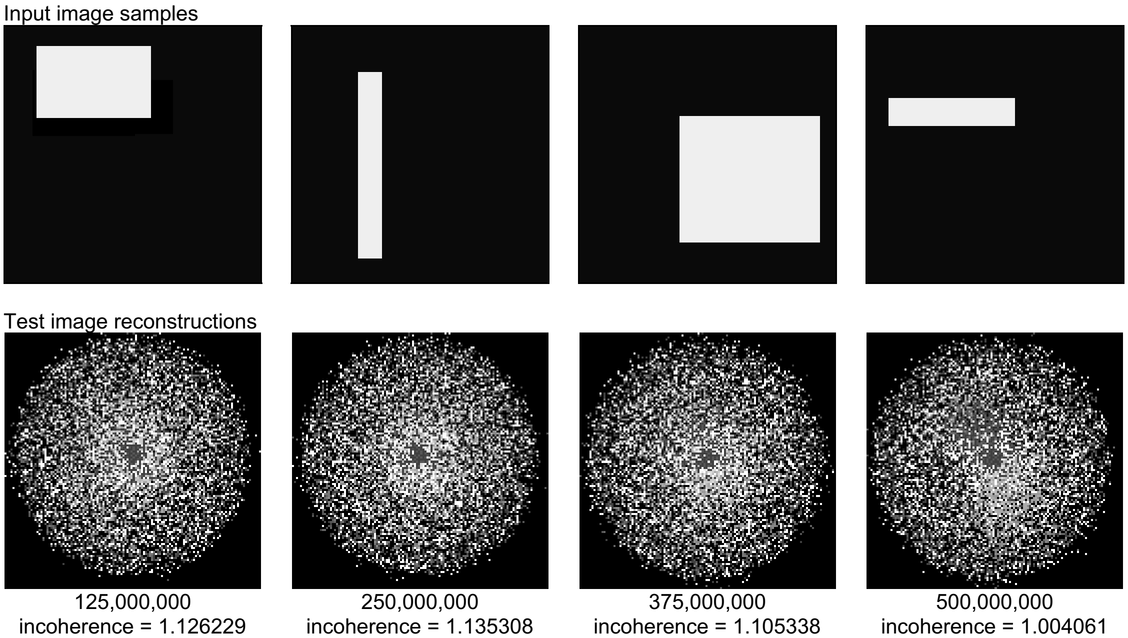
The results are not impressive (see **Figure 42**). Whilst there is visible structure appearing in the geometry after 500 million iterations, and the measure of incoherence is rapidly decreasing, the output picture closely resembles those of **Figure 40**. We suspected the continuing presence of the talking head effect. The edgeness plots do indeed show that μ is reduced for pairs close to the edge of the window.

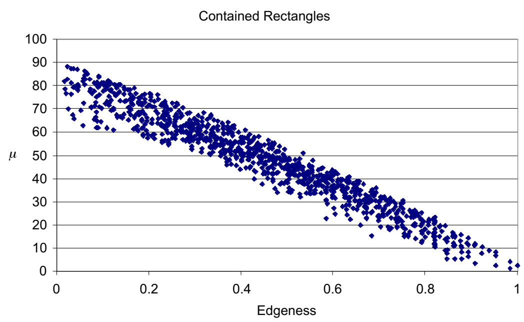
NB: The strange ring effect at 125 million iterations is caused by a group of highly uncorrelated pixels, probably misplaced during the initial run-in of the video. There is often more noise and spurious correlation patterns at the start and end of video tapes.

10.2.4 Experiment D1 – contained rectangles

Our next two experiments (D1 and D2) were intended to isolate the talking head effect in strict visual and mathematical terms. Let us first discuss experiment D1.

A synthesised sequence, in which each frame consisted of a randomly shaped, sized, and positioned rectangle of light colour (grey level 245) appeared against a background of darker colour (grey level 10). The hypothesis was that this sequence would suffer from the talking head effect because there would be considerably greater similarity between opposite edges than between edges and the centre. This is because the rectangles, though random, are all contained within the image window. They are, therefore, as a group, concentrated centrally.





Program Parameters	
image size	128 × 128
sequence	contained rectangles
workforce	complete
agenda	random
iterations per frame	500
descent rate	0.001
estimated μ	255
estimated *a*	-ln (1 – b/μ')
algorithm	1 – a'/g

Figure 43: Contained rectangles.

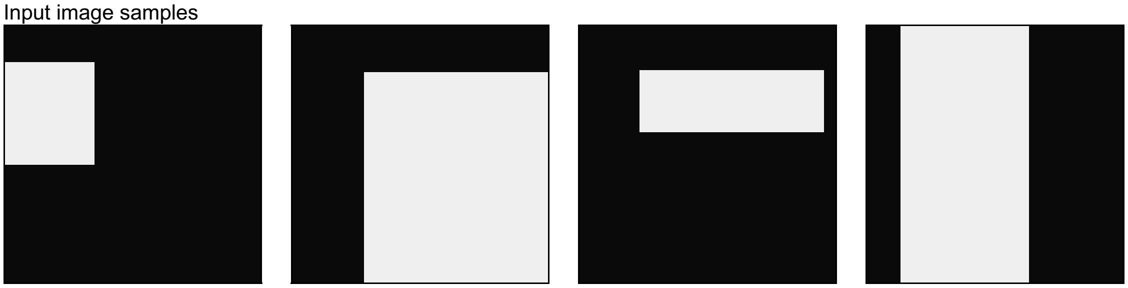
The progress is shown in **Figure 43**. The geometry is again highly everted – even more so than that of experiment D0 (Chomsky video). The plot of μ versus edgeness shows μ dropping rapidly for pairs near the edge of the window. The other half of this experiment-pair was then run.

10.2.5 Experiment D2 – uncontained rectangles

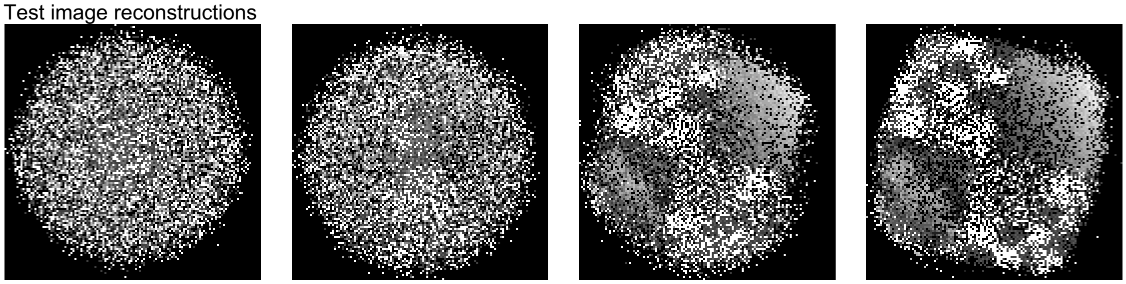
Our hypothesis about the talking head effect states that the common positioning of objects in the centre of images, against a relatively uniform background, will cause misleadingly high similarities between edge areas. In JIGSAW this causes everting, in which the edges of an image move towards each other. This was supported by the experiment D1 with contained rectangles.

In this experiment, a random sequence of rectangles was again created as input, but now each edge of each rectangle had a 50% chance of being outside the window. Consequently, less than 10% of rectangles were totally within the window. All other parameters remained unchanged.

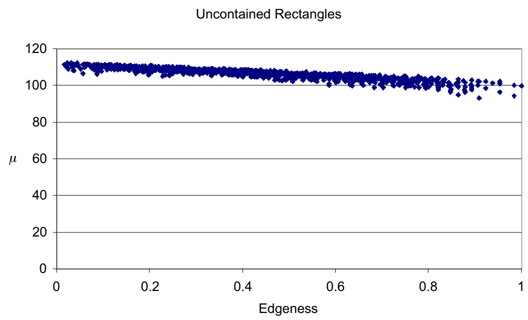
Input image samples



Test image reconstructions



125,000,000	250,000,000	375,000,000	500,000,000
incoherence = 0.618197	incoherence = 0.511162	incoherence = 0.179479	incoherence = 0.039250



Program Parameters

image size	128 × 128
sequence	uncontained rectangles
workforce	complete
agenda	random
iterations per frame	500
descent rate	0.001
estimated μ	255
estimated *a*	-ln (1 – b/μ')
algorithm	1 – a'/g

Figure 44: Uncontained rectangles.

The results are shown in **Figure 44**. The difference between this and the previous experiment is quite marked. Now the geometry is a much better reconstruction of the actual distances, with near-straight edges and no everting. The incoherence measure is much lower, and the edgeness plot gives the reason; there is much more uniform μ across this sequence. Edgeness and μ are only very weakly correlated. This is a strong demonstration of the deleterious nature of the talking head effect.

However, even with most of the talking head effect removed, there are still other generic distortions to be considered. By having no preference for small behavioural differences, many large differences are included in the adjustment of geometry sequence. The inherent tendency to underestimate geometrical distance from large behavioural differences causes a "fish-eye" effect in which the overall image appears to bulge in the middle. The sensor representations are settling into a geometry in which all long g values are disproportionately low.

This effect is aggravated by another effect, which operates at the edges of an image. This has been named the table-cloth effect because it resembles the tendencies of table cloths to fray at the corners. It is suspected that the fraying in this case is due to the predominance of long distances (high a values) selected to position corner and near-corner pixels, as discussed in section 8.4.3.

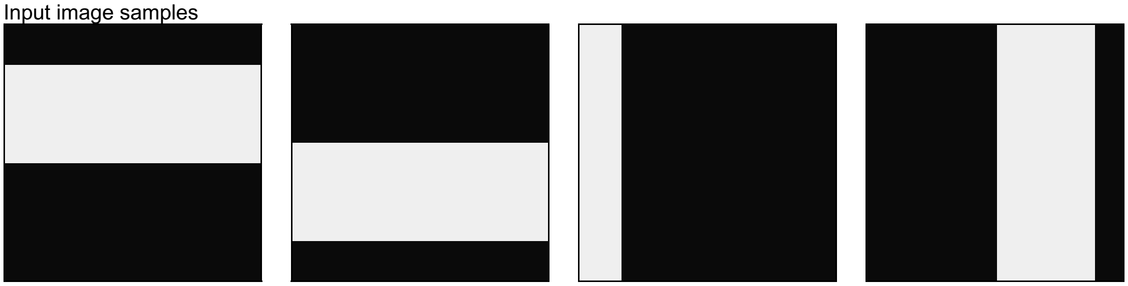
10.2.6 Experiment D3 – activation waves, width = 50

We next investigate which input would be ideal for JIGSAW. A sequence was constructed in which bars moved both horizontally and vertically across the window. Each pixel therefore received the same amount of change, and the same balance of light and dark. Both bars were 50 pixels wide. Again all other parameters were kept constant.

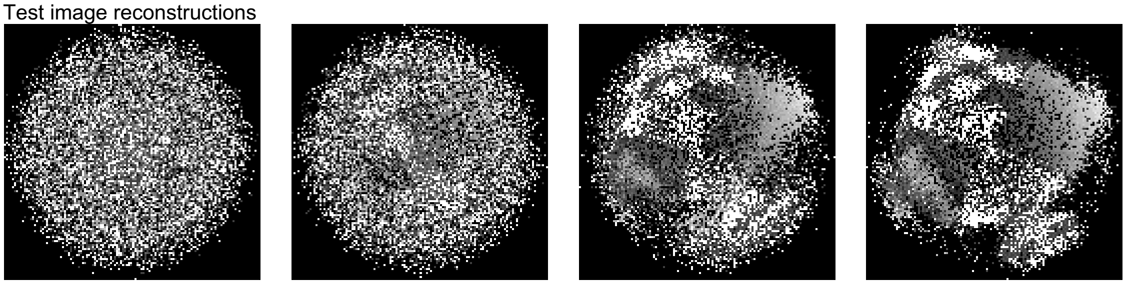
This attempt to create ideal input was inspired by pre-natal retinal activation waves which are thought to be involved in the organisation and adaptation of the retina (Shatz 1992, Feller, Wellis, Stellwagen, Werblin and Shatz 1996). These waves are moving areas of spontaneous activity generated within the retina itself (they occur before birth, in a very dark environment). The phenomenon ceases to operate when a neo-nate's eyes open.

In order to test the hypothesis that JIGSAW-like processes are involved in early retinal organisation, it is first necessary to demonstrate that a sequence of images resembling activation waves will also lead to good reconstruction of space. The sequence consisted of a band of light colour (grey level 245) moving first down, and then across a uniform background of dark colour (grey level 10). This simple sequence was repeated for the whole experiment.

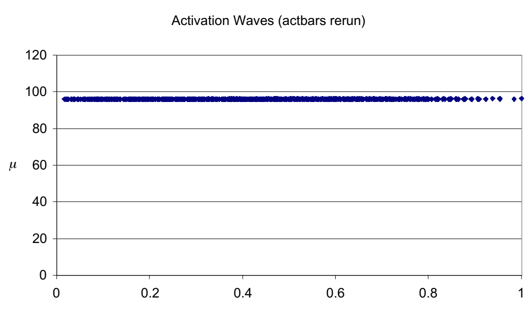
Input image samples



Test image reconstructions



125,000,000	250,000,000	375,000,000	500,000,000
incoherence = 0.615284	incoherence = 0.513438	incoherence = 0.215610	incoherence = 0.088720

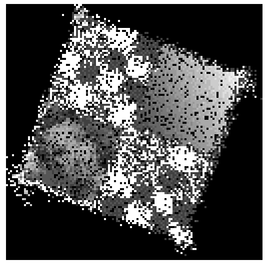


Program Parameters

image size	128 × 128
sequence	activation waves (w = 50)
workforce	complete
agenda	random
iterations per frame	500
descent rate	0.001
estimated μ	255
estimated a	$-\ln(1 - b/\mu')$
algorithm	$1 - a'/g$

Figure 45: Activation waves, width = 50.

In **Figure 45,** the geometry is finding it harder to break away from the circular configurations that form soon after starting. It is also interesting to note that the image at 500 million iterations contains two twists: one at each of the lower corners. Both eventually worked themselves out between 500 million and 625 million iterations (see **Figure 46**).



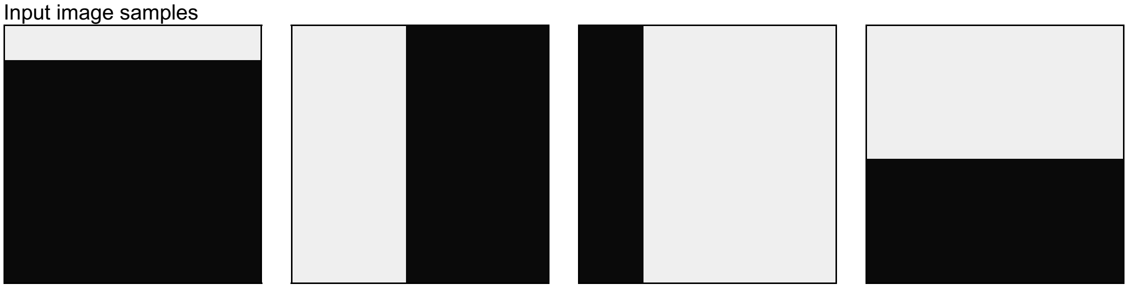
625,000,000
incoherence = 0.011335

Figure 46: Removal of twists

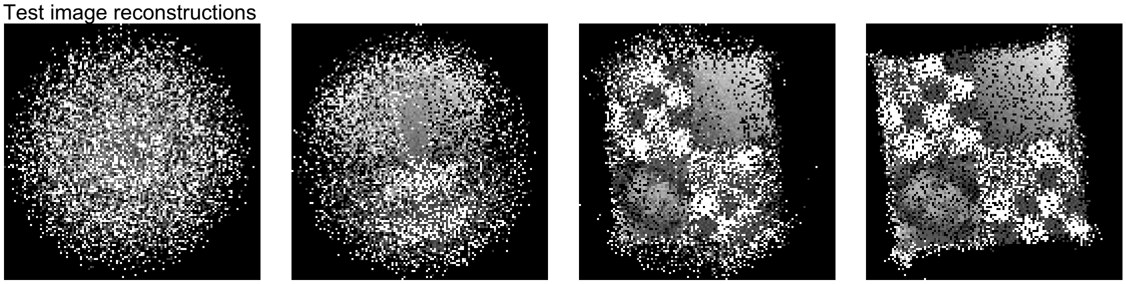
10.2.7 Experiment D4 – activation waves, width = 128

We next created a new sequence in which the bars were exactly the same width as the image (128 pixels). Thus, there could never be an instant at which two areas of like colour were separated by an intervening area of a different colour, as in the case of experiment D3.

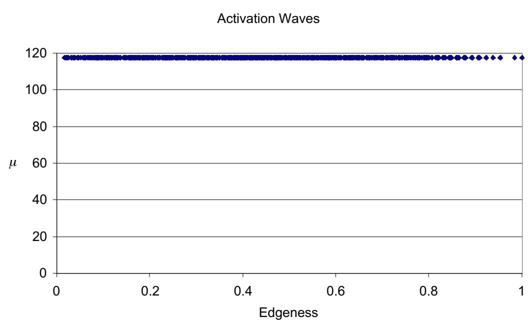
Input image samples



Test image reconstructions



125,000,000	250,000,000	375,000,000	500,000,000
incoherence = 0.579658	incoherence = 0.333468	incoherence = 0.100109	incoherence = 0.011086



Program Parameters

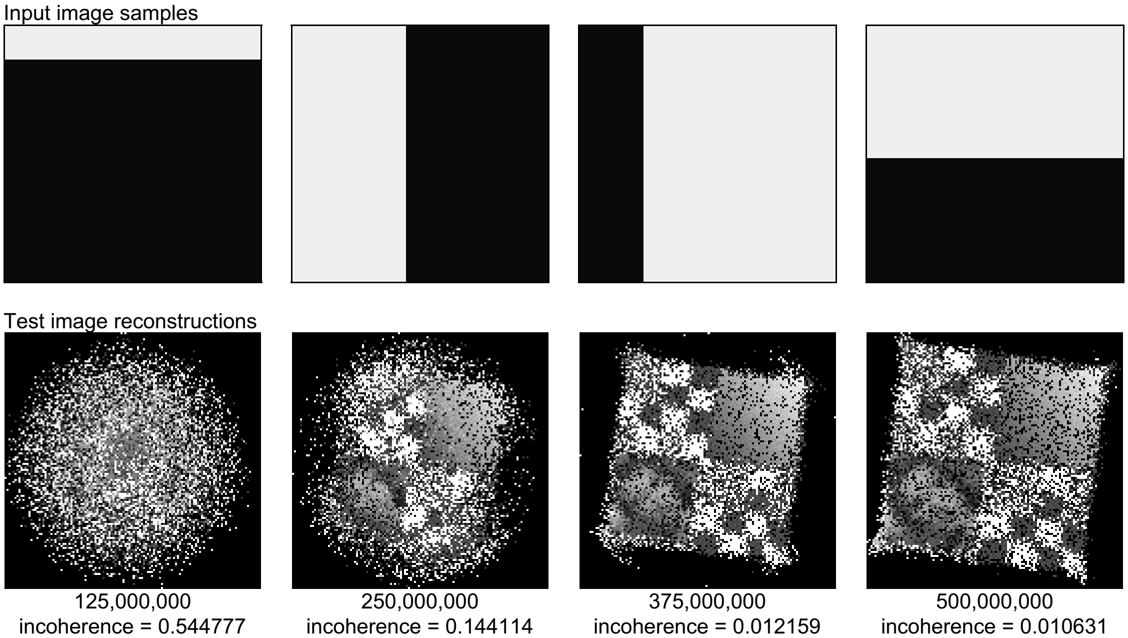
image size	128 × 128
sequence	activation waves (w = 128)
workforce	complete
agenda	random
iterations per frame	500
descent rate	0.001
estimated μ	255
estimated a	-ln (1 – b/μ')
algorithm	1 – a'/g

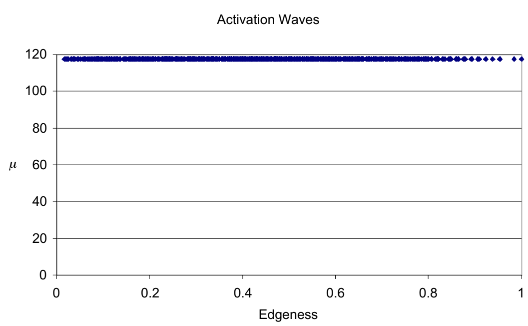
Figure 47: Activation waves, width = 128.

The results are better than those of experiments D0 to D3 (see **Figure 47**), with a very low incoherence measure of 0.011086 after 500 million iterations. There is good reconstruction of the actual image. Edgeness and μ are completely uncorrelated.

10.2.8 Experiment D5 – activation waves, width = 128, linear *ab*

Our analysis of the *ab* curve of this sequence revealed a linear relationship between a and b. In the next experiment we therefore simplified JIGSAW to its crudest possible form, using instantaneous b values as a direct estimate for a, and ran the same sequence of images used in experiment D4.





Program Parameters

image size	128 × 128
sequence	activation waves (w=128)
workforce	complete
agenda	random
iterations per frame	500
descent rate	0.001
estimated μ	255
estimated *a*	*b*
algorithm	1 – a'/g

Figure 48: Activation waves, width = 128.

Figure 48 shows the results – the best so far. The fact that the scene is so well reconstructed, with even the crudest form of the algorithm, is very encouraging. It seems to suggest that activation waves are the ideal input for retinal organisation, and perhaps this is the reason our visual systems have evolved to produce this kind of activity.

We still observe the table-cloth effect in these geometries, and suspect that, as the data is probably optimal, the cause of this lies somewhere in the computation, most probably with pair selection.

10.2.9 Conclusions from experiments with dynamic scenes

The experiments D0 to D5 supplied dynamic video sequences to an austere version of JIGSAW. This version used instantaneous b values, had no workforce sampling or evolution, no b limits, and used a global, constant μ'. Despite this, encouraging results were obtained. The key points from these experiments are:

- Plain television sequences show a high level of talking head effect, and are not effective as input for an austere implementation of JIGSAW. At least two possible solutions to this problem exist. First, if the input were delivered by a camera mounted on a pan-tilt mount, all artistic deliberation could be removed by setting the equipment in a scene of reasonable λ and allowing the pan-tilt head to reposition the camera slightly for each frame. Second, we can accumulate b and compute μ values that are local to work units. Locally computed μ values should protect against the talking head effect.
- Wide activation bars are the ideal input for the crudest form of JIGSAW.
- Other effects remain in the data. The corners of the geometries are not as well correlated with corners of the image, when compared to central sections. This could be due to several reasons: (1) the distribution of distances at corners is less conducive to accurate positioning, (2) corners may suffer from having all forces operate from one side only, (3) the resolution of any twists may give preference to central organisations, at the expense of corners, or (4) some other reason we do not yet know about.

11 Discussion and future work

The purpose of this technical report is to describe a new behaviour-based way of thinking about visual information, and to show that this approach also applies to other domains (such as weather data). This is an important alternative to existing approaches as it can improve the robustness of autonomous systems and provide solutions to the uncontrolled placement of sensors. Increasing autonomy and reducing assumptions about the domain have been strong philosophical principles driving this research. We have set out our assumptions, as clearly as we can see them, in section 2.2. These are simple assumptions revolving around the behavioural, spatial and temporal relationships between sensors.

The mathematical model presented in section 6 describes the relationship between space and behaviour. It has a good fit with the empirical investigations in section 7, but deviations from the idealised theory needed an explanation. There are a number of characteristics of the sensor system which impact on the theory. These include artistic deliberation, the shape of the window and signal interference.

JIGSAW was presented as an algorithm for extracting spatial information from the behaviour of a set of sensor signals. To manage the varying levels of domain knowledge and computational resource available to a JIGSAW implementor, we have provided a number of algorithm variations. In particular, there are methods for implementing the heuristic, “the smaller the b value, the more reliable it is” and various methods for setting the parameter μ'.

At the most philosophical level, it could be argued that space is not only sufficient, but necessary for representation. This is a philosophical assumption that is buried in this research program, but the JIGSAW algorithm is not dependent upon it. The philosophical assumption is that representing collections of phenomena in geometrical spaces is more useful than merely listing differences between phenomena. To put it epigrammatically, a collection of representations is not a representation of collection.

11.1 Performance and efficiency

We established that JIGSAW works on a number of data sets, including many that would confound standard image processing algorithms. We designed and demonstrated special synthesised data sets that enable clear evaluation of techniques. These include synthesised behaviour measures (used in the S series of experiments) and synthesised video input (used in the D series of experiments). These tests enabled us to verify that the basic algorithm is valid, and have shown that unexplained phenomena still exist in real motion data. Without such tests (i.e., using real video input) it is immensely difficult to determine whether phenomena in the results are data-related or algorithm-related. These problems seriously slowed previous research (Peters 2000). The importance of these tests should therefore not be underestimated.

A goal of the algorithm design was to limit the space and time complexity to $O(n)$. This limit has been respected, although more work is needed to understand the speed at which JIGSAW converges on solutions, and the quality of those solutions.

11.2 Neuroscience implications

The major difference between JIGSAW's geometrical space and cortical space is that the first is infinite and the second is not. Because of this, the movements of JIGSAW's sensor representations are independent, yet the movements of axon tips are limited to (1) the spaces between cells and, (2) the finite area of the cortex. This may mean that the modelling of neural processes will have to entail an algorithm that operates more like a tile-shuffling game – so, when a sensor representation is moved to a new position, other representations may have to be moved out of the way.

We believe that this difference is significant, because both retinotopic and homuncular representations are distorted – a disproportionately large area of visual cortex is devoted to foveal input, and the sensory homunculus has relatively large hands and face. These spatial distortions are almost certainly due to the variation in density of sensors (photo-receptors in the retina, and tactile sensors in the skin). Sensory areas of high density are represented by large areas in the brain. This is congruent with the hypothesis that space restrictions have impact in the cortex. To more closely model axonal routing, JIGSAW requires modification. The results of our experiments with 'shuffle' versions of the algorithm will be the subject of a later publication.

However, even with the current form of the algorithm, we have noticed properties with interesting implications for neuroscience. First, activation waves are the source of input most conducive to accurate spatial reconstruction found so far. Second, blurring of the input image is also conducive to good spatial reconstruction. That first activation waves (in the pre-natal phase), and then blurring (in the neo-natal phase) are present in human visual development may be no coincidence.

Our best results were obtained with wide activation bars, moving rapidly across the visual field. However, it appears (Shatz 1992) that activation bars in prenatal retina are quite narrow, and slow-moving. Shatz hypothesises that the slowness may be necessary for activation waves to be detected above the base level of spontaneous neural activity throughout the nascent visual system. It may also be that only when activity is at a sustained high level that axonal repositioning is fully operational. We do not yet know the exact mechanisms of the neural responses involved. There are numerous implications for JIGSAW in these neurobiological observations.

11.3 Applying and extending the method

In cases of extremely remote data collection, it may not be possible to calibrate sensors or even to locate them accurately. We suggest that it is better to be able to overcome calibration problems than to hope that they will not occur. The robustness of JIGSAW in the face of disruption of the sensor array enables this.

If there is frequent sensor adjustment or uncontrollable sensor placement, then JIGSAW could be a useful tool to recover representational validity. This suggests new opportunities in the fields of planetary and deep sea exploration, medical sensing and remote net-based security systems.

At its most speculative, thinking about JIGSAW touches on ideas concerning the creation of completely new knowledge. JIGSAW uses redundancy in data to organise the data, but having done so, there are likely to be circumstances in which the redundancy can be filtered without high risk. It appears that the development of human vision follows such a path. After the establishment of axonal connections is complete in early childhood, the opportunity to repeat the organising process seems to be lost forever. In contrast to adult cataracts, childhood cataracts can result in permanent blindness if they are allowed to interfere with visual input during the critical time of connection-forming processes. It appears that these processes are activated according to a developmental timetable rather than as an on-going response to data. Moreover, redundancy seems to be permanently removed from visual data in normal adult operations, thereby reducing the amount of information to be processed. Vision often appears to operate as a system for alerting the brain to changes, rather than maintaining a constant picture of the outside world (Dong and Atick 1995).

We wonder if the use of redundancy to organise representational frameworks of a higher order, followed by the discarding of it, is a cyclic process producing ever-more stratified yet concise and valid representations of the world. This suggests applying JIGSAW to general machine learning and data-mining domains.

11.4 Integrating JIGSAW into larger systems

If JIGSAW is to be used as a component within larger systems, such as autonomous robots, then it must be interfaced with other functions. We are aware that there already exist vision algorithms that accept stochastically positioned pixels as input. For example, Peters (1998b) has demonstrated that edge detection and motion tracking can be implemented this way. However, this field of research remains relatively untouched.

As robots become more prevalent, and people continue to expect them to become more like self-adjusting humans and less like perseverating machinery, demand for algorithms that are able to work with self-organising sensory representations should increase. There may also be much scope for productive research into methods for integrating dynamic spatial representations with the large body of standard convolution-based techniques already available (e.g., Bederson, Wallace and Schwartz 1995).

11.5 Conclusion

JIGSAW started by examining one of the most fundamental and quietly ubiquitous assumptions in machine vision, that the spatial arrangement of pixels is given. In challenging this assumption we discovered that is it not a necessary foundation on which to build vision systems, rather that signal behaviour is sufficient. In pursuing this challenge we have uncovered an enabling mechanism linking space and behaviour which is far more general than vision.

We believe that the field is now open for further refinement of the techniques, to apply JIGSAW to other domains, to include it into larger autonomous systems, and to use it for understanding natural vision.

12 References

Ackbar, M., & Wintonick, P. (1992). *Manufacturing consent: Noam Chomsky and the media*: National Film Board of Canada.

Arfken, G. B. (1985). The Method of Steepest Descents, *Mathematical methods for physicists* (3rd ed., pp. 428-436). Orlando: Academic Press.

Bederson, B.B., Wallace, R.S., & Schwartz, E. (1995). A miniaturized space-variant active vision system: Cortex-1. *Machine Vision and Applications, 1995*(8), 101-109.

Dong, D.W., & Atick, J.J. (1995). Temporal decorrelation: a theory of lagged and nonlagged responses in the lateral geniculate nucleus. *Network: Computation in Neural Systems, 6*(2), 159-178.

Feller, M.B., Wellis, D.P., Stellwagen, D., Werblin, F.S., & Shatz, C.J. (1996). Cholinergic synaptic transmission is required for wave propagation in developing retina. *Science, 272*, 1182.

Gonzales, R.C., & Woods, R.E. (1993). *Digital image processing*. Reading Massachusetts: Addison-Wesley.

Kirkparick, S., Gelatt, C.D., & Vecchi, M.P. (1983). Optimization by simulated annealing. *Science, 220*(4598), 671-680.

Kolb, B., & Whishaw, I.Q. (1990). *Fundamentals of human neuropsychology* (3rd ed.). New York: W H Freeman.

Kretzmer, E.R. (1952). Statistics of television signals. *Bell System Technology Journal, 31*(4), 751-763.

Mitchison, G. (1991). Neuronal branching patterns and the economy of cortical wiring. *Proceedings of the Royal Society of London, Series B, 245*, 151-158.

Morse, P. M., & Feshbach, H. (1953). Asymptotic series; method of steepest descent, *Methods of theoretical physics, part 1* (pp. 434-443). New York: McGraw-Hill.

Peters, M.W. (1998a). Spatial competence via self-organisation: an intersect of perception and development, *Conference of the Cognitive Science Society* (pp. 830-835), Madison, Wisconsin: Lawrence Erlbaum Associates.

Peters, M. W. (1998b) Integrated techniques for self-organisation, sampling, habituation, and motion-tracking in visual robotics applications, *Machine Vision Applications '98* (pp. 199-202), Makuhari, Japan: IAPR.

Peters, M.W. (2000). *Making sense: autonomy and adaptation in visual robotics*. PhD, University of New South Wales.

Prokopowicz, P.N. (1995). *The development of perceptual integration across eye movements in visual robots*. PhD, Northwestern University.

Shatz, C.J. (1992). The developing brain. *Scientific American* (September), 34-41.

Srinivasan, M.V., Laughlin, S.B., & Dubs, A. (1982). Predictive coding: a fresh view of inhibition in the retina. *Proceedings of the Royal Society of London, Series B, 216*, 427-459.

Zeki, S. (1992). The visual image in mind and brain, *Scientific American* (September), 42-50.

13 Appendix 1 - Glossary of variables and parameters

α	The lowest value obtained from a sensor.
β	The highest value obtained from a sensor.
δ	The measure of incoherence that captures the disparity between the actual distances and geometrical distances of a system of sensors. This is scale-independent, unlike ε.
ε	The total error which captures the disparity between the actual distances and geometrical distances of a system of sensors. This is used to formulate the gradient descent and is scale-dependent. It can be used with subscripts to indicate individual sensor components.
λ	The frequency of region boundaries per unit distance.
λ'	A parameter used in JIGSAW as an estimate of λ.
μ	The expected behavioral difference of two sensors, independent of distance. This can be used with subscripts to identify the sensors.
μ'	A parameter used by JIGSAW as an estimate of μ.
a	The actual distance between two sensors. This is the physical distance between the sensors (or an abstract analogue). It can be used with subscripts to make the sensor explicit.
a'	The inferred actual distance.
b	The behavioral difference between two sensors. This is the opposite of the correlation between two sensor signals. Sometimes this is used to refer to the instantaneous difference between to sensor values. This can be used with subscripts to identify the sensors.
$\hat{b}$	Same as b, but makes explicit that this is an instantaneous value, not an accumulated value.
Erf	The integral of the Gaussian distribution, $\mathrm{Erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\, dt$
f	The function relating b to a, $b = f(a)$.
g	The geometrical difference between two sensors. This distance is a result of representing sensors in a particular geometry. This can be used with subscripts to identify the sensors.
h	The change in a sensor's geometry. This is normally used with subscripts to indicate dimension and sensor(s).
i	A variable which generally ranges over sensors.
j	A variable which generally ranges over sensors.
k	The gradient descent rate. (Also used as a generic scaling constant.)
k_{break}	A parameter controlling the probability of redeploying a unit in an evolved workforce.
k_{hard}	A parameter controlling the hard cut-off value of b for the effectiveness of a unit.
k_{soft}	A parameter controlling the soft cut-off value of b for the effectiveness of a unit.

k_{tries}	A parameter controlling the sample size when redeploying a unit in an evolved workforce.
n	The number of sensors in the system. (Also used as a generic counter.)
P	The probability of an event occurring. The event is represented in a subscript.
p_i	The probability density function for values reported by sensor i.
q_i	A metric to indicate how close a pixel, i, is to the edge of the window.
r	Range of values for a type of sensor.
s_i	Standard deviation for readings from sensor i.
$s_{i,j}$	Standard deviation for the difference between two sensor readings.
t	A variable which generally ranges over time.
v_i	Mean value for readings from sensor i.
$v_{i,j}$	Difference in mean values for a pair of sensors.
w	The number of units in the workforce.
X_i	A vector describing the geometrical position representing a sensor, i.
x_i	One dimension of the vector, X_i, describing the geometrical position representing a sensor, i.

14 Appendix 2 - Derivation of μ for uniform discrete distributions

The problem is to find the expected difference in sensor readings, between a pair of discrete sensors, *i* and *j*, given that the readings are uniformly distributed. The distribution for sensor *i* is from α_i to β_i and for sensor *j* is α_j to β_j. The expected difference is therefore

$$\begin{aligned}\mu_{i,j} &= \sum_{x=\alpha_i}^{\beta_i}\sum_{y=\alpha_j}^{\beta_j}\frac{1}{\left(1+\beta_i-\alpha_i\right)}\frac{1}{\left(1+\beta_j-\alpha_j\right)}\left|x-y\right| \\ &= \frac{1}{\left(1+\beta_i-\alpha_i\right)\left(1+\beta_j-\alpha_j\right)}\sum_{x=\alpha_i}^{\beta_i}\sum_{y=\alpha_j}^{\beta_j}\left|x-y\right|\end{aligned} \tag{44}$$

In order to find a closed form solution to the double sum , we first define a simpler version of it where the range of each sum is from 0 to α. Call this simpler double sum, $\Diamond(\alpha)$, which is defined by

$$\Diamond\left(\alpha\right) = \sum_{x=0}^{\alpha}\sum_{y=0}^{\alpha}\left|x-y\right| \tag{45}$$

It is not difficult to show that $\Diamond(\alpha)$ has the closed form solution

$$\Diamond\left(\alpha\right) = \frac{\left(\alpha+1\right)^3-\alpha-1}{3} \tag{46}$$

Now consider the table of the function $|x - y|$, a portion of which is shown in **Table 4**.

	0	1	2	3	4	5
0	0	1	2	3	4	5
1	1	0	1	2	3	4
2	2	1	0	1	2	3
3	3	2	1	0	1	2
4	4	3	2	1	0	1
5	5	4	3	2	1	0

Table 4: A portion of tabulated results of the function $|x - y|$.

By considering this table, it can be seen that adding the contents of a rectangle with one corner at (0, 0) and the other at (α_x, α_y), represents a double sum, which we define as

$$\Diamond\left(\alpha_x,\alpha_y\right) = \sum_{x=0}^{\alpha_x}\sum_{y=0}^{\alpha_y}\left|x-y\right| \tag{47}$$

Furthermore, the rectangle representing $\Diamond(\alpha_x, \alpha_y)$ can be constructed from square portions. (There is a similarity with the algebraic truth: $2\,a\,b = a^2 + b^2 - (a-b)^2$.) This gives a closed form solution to $\Diamond(\alpha_x, \alpha_y)$ based on $\Diamond(\alpha)$, which is

$$\Diamond(\alpha_x, \alpha_y) = \frac{\Diamond(\alpha_x) + \Diamond(\alpha_y) - \Diamond(|\alpha_x - \alpha_y| - 1)}{2} \tag{48}$$

Similarly, the original double sum can be formulated using the newly defined rectangles, giving the solution

$$\sum_{x=\alpha_x}^{\beta_x} \sum_{y=\alpha_y}^{\beta_y} |x - y| = \Diamond(\beta_x, \beta_y) - \Diamond(\alpha_x - 1, \beta_y) - \Diamond(\beta_x, \alpha_y - 1) + \Diamond(\alpha_x - 1, \alpha_y - 1) \tag{49}$$

By expanding the definitions of $\Diamond$ the complete solution is

$$\begin{aligned} \mu_{i,j} &= \sum_{x=\alpha_i}^{\beta_i} \sum_{y=\alpha_j}^{\beta_j} \frac{1}{(1 + \beta_i - \alpha_i)} \frac{1}{(1 + \beta_j - \alpha_j)} |x - y| \\ &= \frac{\begin{pmatrix} |\alpha_i - \alpha_j| - |\alpha_i - \alpha_j|^3 + |\beta_i - \beta_j| - |\beta_i - \beta_j|^3 \\ + \alpha_i + \alpha_j - \beta_i - \beta_j + (1 - \alpha_i + \beta_j)^3 + (1 - \alpha_j + \beta_i)^3 - 2 \end{pmatrix}}{6(\alpha_i - \beta_i - 1)(\alpha_j - \beta_j - 1)} \end{aligned} \tag{50}$$

15 Appendix 3 - Derivation of μ for uniform continuous distributions

The problem is to find the expected difference in sensor readings, between a pair of sensors, *i* and *j*, given that the readings form a continuous, uniform distribution. The distribution for sensor *i* is from α_i to β_i and for sensor *j* is α_j to β_j. The expected difference is therefore

$$\begin{aligned}\mu_{i,j} &= \int_{\alpha_i}^{\beta_i}\int_{\alpha_j}^{\beta_j} \frac{1}{(\beta_i-\alpha_i)}\frac{1}{(\beta_j-\alpha_j)}|x-y|\,dx\,dy \\ &= \frac{1}{(\beta_i-\alpha_i)(\beta_j-\alpha_j)}\int_{\alpha_i}^{\beta_i}\int_{\alpha_j}^{\beta_j}|x-y|\,dx\,dy\end{aligned} \tag{51}$$

In similar fashion to the method used in appendix 2, the volume under the rectangular surface for the function $|x-y|$ can be constructed from volumes under squares. For the volume under the square from (0, 0) to (α, α) we have the solution

$$\begin{aligned}\Diamond(\alpha) &= \int_0^\alpha\int_0^\alpha |x-y|\,dx\,dy \\ &= \frac{\alpha}{3}\end{aligned} \tag{52}$$

The same reasoning used in appendix 2 (exploiting symmetries in the function $|x-y|$) allow $\Diamond(\alpha)$ to be used to define the more complicated volume under the rectangle (0, 0) to (α_x, α_y). This results in

$$\begin{aligned}\Diamond(\alpha_x,\alpha_y) &= \int_0^{\alpha_x}\int_0^{\alpha_y}|x-y|\,dx\,dy \\ &= \frac{\Diamond(\alpha_x)+\Diamond(\alpha_y)-\Diamond(|\alpha_x-\alpha_y|)}{2}\end{aligned} \tag{53}$$

Similarly, the original double-integral can be formulated using the newly defined rectangles, giving the following solution

$$\int_{\alpha_x}^{\beta_x}\int_{\alpha_y}^{\beta_y}|x-y|\,dx\,dy = \Diamond(\beta_x,\beta_y)-\Diamond(\alpha_x,\beta_y)-\Diamond(\beta_x,\alpha_y)+\Diamond(\alpha_x,\alpha_y) \tag{54}$$

By expanding the definitions of $\Diamond$ the complete solution is

$$\begin{aligned}\mu_{i,j} &= \int_{\alpha_i}^{\beta_i}\int_{\alpha_j}^{\beta_j} \frac{1}{(\beta_i-\alpha_i)}\frac{1}{(\beta_j-\alpha_j)}|x-y|\,dx\,dy \\ &= \frac{3\left(\alpha_i\alpha_j+\beta_i\beta_j+\alpha_i\beta_j+\alpha_j\beta_i-\alpha_j^2-\beta_j^2\right)-2\left(\alpha_i^2+\beta_i^2+\alpha_i\beta_i\right)}{6(\alpha_j-\beta_j)}\end{aligned} \tag{55}$$

16 Appendix 4 - Derivation of μ for normal distributions

The problem is to find the expected difference in sensor readings between a pair of sensors, given that the readings are normally distributed.

Assume the sensor, i, has values normally distributed with mean, v_i, and standard deviation, s_i. The probability of obtaining various readings at sensor i is given by the normal distribution probability density function,

$$p_i(x) = \frac{e^{-\frac{(x-v_i)^2}{2s_i^2}}}{s_i\sqrt{2\pi}} \tag{56}$$

The expected difference between a pair of sensors, i and j, can be estimated by

$$\mu_{i,j} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p_i(x)\,p_j(y)\,|x-y|\;dy\,dx \tag{57}$$

By considering the z-scores of integration variables, x and y, this can be simplified to

$$\mu_{i,j} = \tfrac{1}{2\pi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-\frac{1}{2}\left(x^2+y^2\right)}\left|\left(v_i + s_i x\right) - \left(v_j + s_j y\right)\right|\;dy\,dx \tag{58}$$

Symmetries in the problem mean that this can be further simplified by considering the expected difference of one theoretical sensor from a zero point. That theoretical sensor would have mean value, $v_{i,j}$, and standard deviation, $s_{i,j}$, given by

$$v_{i,j} = \left|v_i - v_j\right| \tag{59}$$

$$s_{i,j} = \sqrt{s_i^2 + s_j^2} \tag{60}$$

Therefore $\mu_{i,j}$ can be calculated using

$$\mu_{i,j} = \tfrac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} e^{-\frac{1}{2}x^2}\left|v_{i,j} + s_{i,j}x\right|\;dx \tag{61}$$

To remove the absolute function we note that the integrated function will cross the x axis when $s_{i,j}x$ equals v. This implies that the integral can be divided into two terms without the absolute function, giving

$$\tfrac{1}{\sqrt{2\pi}}\left[\left(\int_{-v_{i,j}/s_{i,j}}^{\infty} e^{-\frac{1}{2}x^2}\left(v_{i,j} + s_{i,j}x\right)dx\right) - \left(\int_{-\infty}^{-v_{i,j}/s_{i,j}} e^{-\frac{1}{2}x^2}\left(v_{i,j} + s_{i,j}x\right)dx\right)\right] \tag{62}$$

To prepare for further simplification, each integral can be decomposed into the sum of two other integrals:

$$\tfrac{1}{\sqrt{2\pi}}\left[\begin{array}{l}\left(\int_{0}^{\infty} e^{-\frac{1}{2}x^2}\left(v_{i,j} + s_{i,j}x\right)dx\right) + \left(\int_{-v_{i,j}/s_{i,j}}^{0} e^{-\frac{1}{2}x^2}\left(v_{i,j} + s_{i,j}x\right)dx\right) \\ -\left(\int_{-\infty}^{0} e^{-\frac{1}{2}x^2}\left(v_{i,j} + s_{i,j}x\right)dx\right) + \left(\int_{-v_{i,j}/s_{i,j}}^{0} e^{-\frac{1}{2}x^2}\left(v_{i,j} + s_{i,j}x\right)dx\right)\end{array}\right] \tag{63}$$

and as the second and last terms are equal, this simplifies to

$$\tfrac{1}{\sqrt{2\pi}}\left[\left(\int_0^\infty e^{-\frac{1}{2}x^2}\left(v_{i,j}+s_{i,j}x\right)dx\right)-\left(\int_{-\infty}^0 e^{-\frac{1}{2}x^2}\left(v_{i,j}+s_{i,j}x\right)dx\right)+2\left(\int_{-v_{i,j}/s_{i,j}}^0 e^{-\frac{1}{2}x^2}\left(v_{i,j}+s_{i,j}x\right)dx\right)\right] \tag{64}$$

The first and second terms can be expanded which, when rearranged, produce

$$\tfrac{1}{\sqrt{2\pi}}\left[\begin{array}{l}\left(\int_0^\infty e^{-\frac{1}{2}x^2}v_{i,j}\,dx\right)-\left(\int_{-\infty}^0 e^{-\frac{1}{2}x^2}v_{i,j}\,dx\right)+\left(\int_0^\infty e^{-\frac{1}{2}x^2}s_{i,j}x\,dx\right)-\left(\int_{-\infty}^0 e^{-\frac{1}{2}x^2}s_{i,j}x\,dx\right)\\ +2\left(\int_{-v_{i,j}/s_{i,j}}^0 e^{-\frac{1}{2}x^2}\left(v_{i,j}+s_{i,j}x\right)dx\right)\end{array}\right] \tag{65}$$

Due to the symmetry of the integrated functions about the *y* axis, the first two terms cancel out, leaving

$$\tfrac{1}{\sqrt{2\pi}}\left[\left(\int_0^\infty e^{-\frac{1}{2}x^2}s_{i,j}x\,dx\right)-\left(\int_{-\infty}^0 e^{-\frac{1}{2}x^2}s_{i,j}x\,dx\right)+2\left(\int_{-v_{i,j}/s_{i,j}}^0 e^{-\frac{1}{2}x^2}\left(v_{i,j}+s_{i,j}x\right)dx\right)\right] \tag{66}$$

In a similar fashion, the next two terms can be combined, giving

$$\tfrac{1}{\sqrt{2\pi}}\left[2\left(\int_0^\infty e^{-\frac{1}{2}x^2}s_{i,j}x\,dx\right)+2\left(\int_{-v_{i,j}/s_{i,j}}^0 e^{-\frac{1}{2}x^2}\left(v_{i,j}+s_{i,j}x\right)dx\right)\right] \tag{67}$$

which simplifies to

$$\tfrac{2}{\sqrt{2\pi}}s_{i,j}\left[\left(\int_0^\infty e^{-\frac{1}{2}x^2}x\,dx\right)+\left(\int_{-v_{i,j}/s_{i,j}}^0 e^{-\frac{1}{2}x^2}\left(\frac{v_{i,j}}{s_{i,j}}+x\right)dx\right)\right] \tag{68}$$

The first integral reduces to 1 and the second integral can be rewritten, giving

$$\tfrac{2}{\sqrt{2\pi}}s_{i,j}\left[1+\int_0^{v_{i,j}/s_{i,j}} e^{-\frac{1}{2}x^2}\left(\frac{v_{i,j}}{s_{i,j}}-x\right)dx\right] \tag{69}$$

This is equivalent to

$$\tfrac{2}{\sqrt{2\pi}}s_{i,j}\left[e^{-\frac{1}{2}\left(\frac{v_{i,j}}{s_{i,j}}\right)^2}+\sqrt{\tfrac{\pi}{2}}\,\tfrac{v_{i,j}}{s_{i,j}}\,\mathrm{Erf}\left(\tfrac{1}{\sqrt{2}}\tfrac{v_{i,j}}{s_{i,j}}\right)\right] \tag{70}$$

where Erf(x) is the error function defined as the integral of the Gaussian Distribution,

$$\mathrm{Erf}\left(x\right)=\frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}\,dt \tag{71}$$

Continuing with the simplification, we have

$$
\begin{aligned}
\mu_{i,j} &= \frac{2}{\sqrt{2\pi}} s_{i,j} \left[e^{-\frac{1}{2}\left(\frac{v_{i,j}}{s_{i,j}}\right)^2} + \sqrt{\frac{\pi}{2}} \frac{v_{i,j}}{s_{i,j}} \operatorname{Erf}\left(\frac{1}{\sqrt{2}} \frac{v_{i,j}}{s_{i,j}} \right) \right] \\
&= \frac{2}{\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{v_{i,j}}{s_{i,j}}\right)^2} s_{i,j} + \frac{2}{\sqrt{2\pi}} s_{i,j} \sqrt{\frac{\pi}{2}} \frac{v_{i,j}}{s_{i,j}} \operatorname{Erf}\left(\frac{1}{\sqrt{2}} \frac{v_{i,j}}{s_{i,j}} \right) \\
&= \sqrt{\frac{2}{\pi}} e^{-\frac{1}{2}\left(\frac{v_{i,j}}{s_{i,j}}\right)^2} s_{i,j} + \operatorname{Erf}\left(\frac{1}{\sqrt{2}} \frac{v_{i,j}}{s_{i,j}} \right) v_{i,j}
\end{aligned}
\tag{72}
$$

17 Appendix 5 - Demonstration of Erf approximation

The Erf function is defined as the integral of the Gaussian distribution,

$$\mathrm{Erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}\, dt \tag{73}$$

We do not know of any simple solution to the Erf integral but note that it can be approximated by

$$\mathrm{Erf}(x) \approx 1 - e^{-\sqrt{2}x} \tag{74}$$

This approximation can be seen graphically by viewing plots of the two functions as shown in **Figure 49**.

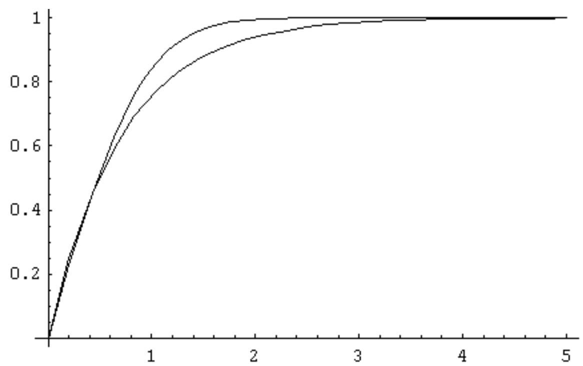


Figure 49: Plot of the Erf function and the approximation, $1 - e^{-\sqrt{2}x}$.

From inspecting the plot we notice that the approximation overestimates for low values of x and underestimates for high values. By looking at a plot of the error in approximation, we can visually place bounds on this error.

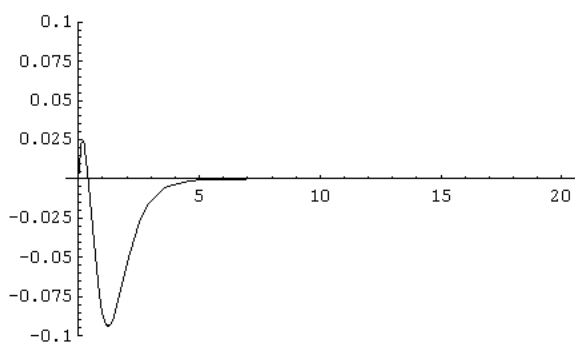


Figure 50: Plot of the error in the approximation, $\left(1 - e^{-\sqrt{2}x}\right) - \mathrm{Erf}(x)$.

It can be seen from **Figure 50** that the approximation to Erf does not overestimate by more than 0.025 nor underestimate by more than 0.095. These bounds are guaranteed beyond the maximum value of x shown in the figure as both functions are monotonically increasing, and have an asymptote of 1. Therefore the maximum error for larger values of x can be no greater than the function's difference from the asymptote at that point (which is much less than 0.025).