# Occupancy Array-Based Kinematic Reasoning

Patrick Olivier, Andrew Ormsby and Keiichi Nakata
Department of Computer Science University of Wales
Aberystwyth Dyfed SY23 3DB United Kingdom
phone +44-1970-622424 fax +44-1970-622455
email plo,aro,kkn@aber.ac.uk

**Abstract:** This paper presents a technique for kinematic reasoning that is based on the use of occupancy arrays. We describe the algorithm used and a prototype implementation which can reason about the behaviour of higher pairs, that is, parts such as cams and gears, with rotational or translational degrees of freedom. We show that by representing the spatial occupancy of objects using an appropriately high resolution occupancy arrays we can reason about higher pair kinematic interactions between objects. Both free and blocked motion can be inferred and we report an implementation, *KAP*, that demonstrates the utility of this approach. abstract

## 1 Introduction

This paper presents a technique for kinematic reasoning that is based on the use of occupancy arrays. We describe the algorithm used and a prototype implementation which can reason about the behaviour of higher pairs, that is, parts such as cams and gears, with rotational or translational degrees of freedom.

Section 2 introduces the representation used. Section 3 provides a detailed account of the algorithm, explaining how the normal at the surface of an object is calculated, how blocked and propagated motions can be inferred, and the importance of the granularity of the occupancy array.

*KAP* uses multiple occupancy arrays at various levels of granularity. An occupancy array at a coarse level of detail is used to identify possible areas of interaction. Section 4 discusses the way in which successively more detailed occupancy arrays are used to focus on the points of contact in an efficient manner.

Section 5 briefly relates the work presented here to qualitative kinematics, analogical reasoning and previous research into spatial reasoning using occupancy arrays. Section 6 identifies some of the key issues for future work.

## 2 Occupancy Arrays

Occupancy arrays are an analogical representation of the spatial properties of objects. An object's spatial occupancy in the world at some location is represented by a corresponding occupancy of elements in the array. The granularity of the array defines the mapping between regions in the world and the indices of the array.

The world occupancy array $W$ represents the spatial region in which we are to perform kinematic reasoning. Each element may be either empty, corresponding to unoccupied space, or contain an identifier for an object indicating the object's spatial occupancy of the corresponding region in the world. The constraint that an element should never be occupied by more than one object is the basis used in inferring interactions between objects.

### 2.1 Component Information

Each component's spatial occupancy is represented using a unique occupancy array. We explain in Section 3 how our approach to kinematic reasoning, and in particular the calculation of the normal to a point on the boundary of a component, requires a distinction to be made between the object boundary and its immediate interior. The representation of the spatial occupancy of further interior elements is not necessary — kinematics is inherently a surface phenomenon. Figure 1 shows a fragment of the component occupancy array for the cam of the cam-follower mechanism discussed in Section 4. The edge representation is generated from the fully occupied array, an element being considered to be on the boundary of an object if it is occupied but one element of its enclosing rectangle is unoccupied. Similarly, an element is considered to be part of the immediate inside region if it is occupied and it is not on the boundary, but an element of its enclosing rectangle is a boundary element.

The remaining spatial properties of the components are represented by the following attributes:

- *Object identifier*
  A unique integer identifying a particular instance of a component.
- *Object type*
  The type of degree of freedom, either *rotational* or *translational.*
- *Degree of freedom*
  A vector describing the degree of freedom, note

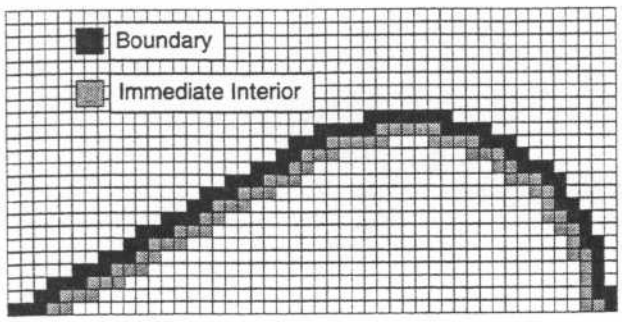


Figure 1: Boundary and interior occupancy array elements.

that for rotational case this is simply {0,0,1}. For rotational objects the position within the component occupancy array of the centre of rotation is also included.

- *Displacement increment*
  This is the unit of motion for the component. For objects with translational degrees of freedom this will be a sub-elemental displacement and for those with a rotational degree of freedom some small angle which will give rise to a sub-elemental displacement of the elements most distant from the axis of the degree of freedom. This issue of increment magnitude is discussed in Section 3.5.

- *Displacement*
  The current displacement or rotation of a component. This is the sum of all previous displacement increments.

- *World location*
  The initial position of a reference point in the component (usually the bottom left corner of the occupancy array) in the world.

- *Dimension*
  Integer extent of the square occupancy array of the component.

- *Granularity*
  The number of occupancy array elements per unit of length.

### 2.2 Object Transformations

Objects in $W$ can be rotated or translated as a consequence of either a driving motion or as a result of an interaction with another object. These transformations may be achieved by standard bitmap rotation and translation operations. The spatial occupancy of the transformed object is computed by inverting the transformation, iterating over the destination occupancy array, applying the inverse transformation to find which source occupancy array element to copy.

Davis [1990] notes a widely held criticism of occupancy arrays that under non-orthogonal translations and rotations there is an information loss, or blurring of the occupancy array edge, of approximately one element per transformation. We avoid this by accumulating the extent of previous translations and rotations in the displacement slot of object frame and each time performing the cumulative transformation with respect to its untransformed occupancy in $W$. Whilst this does not eliminate the one element (per boundary element) information loss for the transformation, the error does not accumulate.

## 3 Kinematic Reasoning

Two components $C_1$ and $C_2$ comprise a higher pair. When $C_1$ is being driven and it interacts spatially with $C_2$, the resulting motion can be inferred from the direction of the normal $\hat{n}$ to the surface of $C_2$ at the point of contact between the two components. In kinematics it is assumed that force can only be transmitted from $C_1$ to $C_2$ in a direction negative to this normal.

If $C_2$ has a translational degree of freedom in direction $\hat{x}$ then the motion of $C_2$ can be inferred from the sign of $(-\hat{n} \cdot \hat{x})$. If the scalar product is positive then $C_1$ causes $C_2$ to move in the direction $\hat{x}$; if negative then $C_2$ moves in the $-\hat{x}$ direction. If the product is zero then further motion of $C_1$ is blocked by $C_2$.

If $C_2$ has a rotational degree of freedom $\hat{\theta}$, and $\vec{r}$ is a vector from the axis of rotation to the point of contact between $C_1$ and $C_2$, then the motion of $C_2$ can be inferred from the sign of $((\vec{r} \times -\hat{n}) \cdot \hat{\theta})$. If the product is positive then the rotation of $C_2$ will be with the sense $\hat{\theta}$; if the result is negative then the sense of the rotation is $-\hat{\theta}$.

### 3.1 Algorithm

We showed in Section 2.1 that the vector representations of the component degrees of freedom are given as part of the object description. As the above explanation indicates, the additional crucial factors in inferring the kinematic behaviour of a higher pair are the identification of the point of interaction between two components $C_1$ and $C_2$, and the calculation of the normal to the surface of $C_2$. We use two occupancy arrays to infer the kinematic behaviour of the pair: the initial world occupancy array $W$ containing the component arrays in their initial configurations, and a temporary array $T$ constructed by transforming the components in $W$ according to the following algorithm:

1. Increment the displacement of $C_1$.

2. Construct $T$ from $W$ by transforming both the driving component $C_1$ through its current displacement and the driven component $C_2$ through its current displacement (initially zero) of translation or rotation depending on their respective degrees of freedom ($\hat{d}_{C1}$ and $\hat{d}_{C2}$).

3. Scan $T$ for elements co-occupied by objects $C_1$ and $C_2$. If such a location exists, then:

   (a) Calculate the normal $\vec{n}$ to the boundary of $C_2$ at the point of contact.

   (b) If $\hat{d}_{C2}$ is translational, then:

      i. If $(-\vec{n} \cdot \hat{d}_{C2})$ is positive, then delete the occupancy array for $C_2$ in $T$, and replace it with the occupancy array in $W$ transformed through its current translational displacement incremented by one. Increment the value of the displacement slot of $C_2$ and return to the beginning of step 3.

      ii. If $(-\vec{n} \cdot \hat{d}_{C2})$ is negative, then delete the occupancy array for $C_2$ in $T$, and replace it with the array in $W$ transformed through its current translational displacement decremented by one. Decrement the value of the displacement slot of $C_2$ and return to the beginning of step 3.

      iii. If $(-\vec{n} \cdot \hat{d}_{C2})$ is zero, then exit noting that the motion of $C_1$ is blocked.

   (c) If $\hat{d}_{C2}$ is rotational, and $\vec{r}$ is the displacement of the point of contact between $C_1$ and $C_2$ from the centre of rotation of $C_2$, then:

      i. If $((\vec{r} \times -\vec{n}) \cdot \hat{d}_{C2})$ is positive, then delete the occupancy array for $C_2$ in $T$, and replace it with the occupancy array for $C_2$ in $W$ transformed through its current rotational displacement incremented by one. Increment the value of the displacement slot of $C_2$ and return to the beginning of step 3.

      ii. If $((\vec{r} \times -\vec{n}) \cdot \hat{d}_{C2})$ is negative, then delete the occupancy array for $C_2$ in $T$, and replace it with the occupancy array for $C_2$ in $W$ transformed through its current rotational displacement decremented by one. Decrement the value of the displacement slot of $C_2$ and return to the beginning of step 3.

      iii. If $((\vec{r} \times -\vec{n}) \cdot \hat{d}_{C2})$ is zero, then exit noting that the motion of $C_1$ is blocked.

4. Return to step 2.

### 3.2 Normal Calculation

One of the fundamental advantages of using a *high granularity* occupancy array over lower resolution grid-like approaches (see Narayanan [1991], Glasgow [1993] and Funt[1980]) is that geometric knowledge such as the location of points on the boundary of an object is preserved to a known degree. A consequence of this is that the gradient and normal at a particular boundary element can be approximated on the basis of the local context, that is, the locations of neighbouring boundary and immediate interior elements.

Consider figure 2. Let boundary element $e_i$ be a point of overlap between components $C_1$ and $C_2$. The direction of the normal at $e_i$ can be approximated as follows:

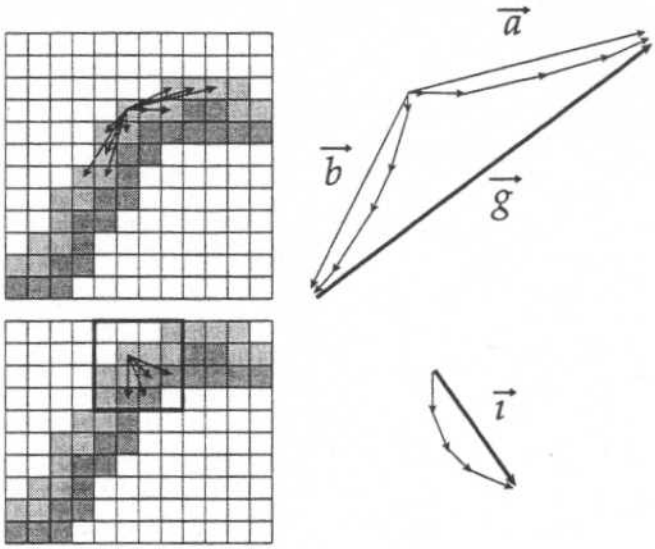


Figure 2: Calculation of the normal.

1. Form two sets of elements $S_a = \{e_{i+1}..e_{i+n}\}$ and $S_b = \{e_{i-1}..e_{i-n}\}$, where each set contains $n$ elements in one of the two directions along the boundary of $C_2$ from $e_i$.

2. For each set sum the vector displacements of each element from $e_i$ to give two vectors $\vec{a}$ and $\vec{b}$.

3. Assign the vector difference, $\vec{a} - \vec{b}$, to $\vec{g}$, so that $\vec{g}$ has a direction approximately tangential to $C_2$ at the point of contact $e_i$.

4. From $\vec{g}$ form an orthogonal vector $\vec{m}$.

5. Check which direction from $e_i$ is the inside of $C_2$. Form a set of elements $S_c$ comprising the immediate interior elements within an uniform enclosing rectangle around $e_i$.

6. Sum the displacements from $e_i$ to give $\vec{i}$.

7. If $(\vec{m} \cdot \vec{i})$ is positive then a non-unit normal at $e_i$ is given by $\vec{n} = -\vec{m}$. Otherwise, $\vec{n} = \vec{m}$.

While we offer no formal justification of the appropriateness of local search in estimating normals to the boundary of occupancy arrays, we observe that the gradient at a point is a property of the cumulative spatial distribution of the surface around the point and not of the point itself.

### 3.3 Blocking

There are two principal mechanisms by which blocking of the driving motion can be affected. Firstly, as indicated by steps 3(b)iii and 3(c)iii of the algorithm in Section 3.1, if the normal of the component being driven has no component in the direction of a translational degree of freedom, or passes through the axis of a rotational degree of freedom of the interacting component, then motion cannot be transmitted.

The other situation where blocking occurs is when two distinct regions of the same driving component simultaneously force opposing motions in the component it interacts with. A typical case of this is when two gears mesh too tightly. Under these conditions the algorithm in Section 3.1 will loop infinitely in its attempt to satisfy the constraint of single object occupancy of array elements. The algorithm can be modified to capture such cases by including an additional exit clause when any object is driven in more than one direction within the same increment of the driving component.

### 3.4 Motion Propagation

In real mechanisms kinematic chains typically consist of multiple pairs. The occupancy array algorithm in Section 3.1 can be simply modified to cater for multiple objects by checking at step 3 for multiple occupancy of elements with respect to any pair of objects. Consider the three gears $C_1$, $C_2$ and $C_3$ of figure 3. When clockwise rotation of $C_1$ (the bottom right gear) takes place this gives rise to a brief period of free movement until an overlap with $C_2$. $C_1$ and $C_2$ then rotate together until $C_2$ eventually interacts with $C_3$.

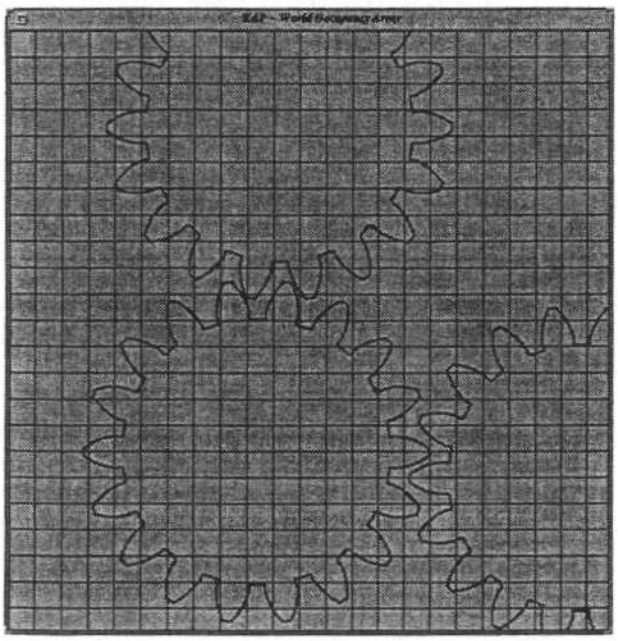

Figure 3: Occupancy array for three interacting gears.

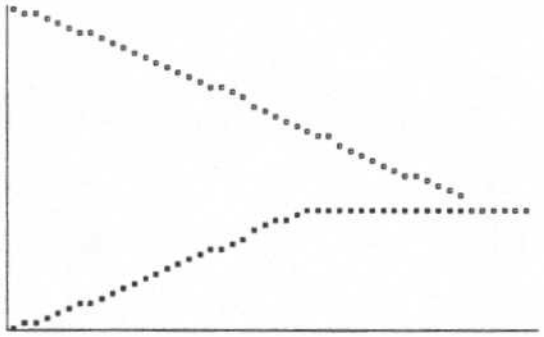

Figure 4: Simulation results for three gears. The horizontal axis represents the angular displacement of gear $C_1$ from right to left. Light points (upper track) indicate the angular displacement of $C_2$. Dark points (lower track) indicate the angular displacement of $C_3$.

### 3.5 Granularity and Transformation Increments

The choice of granularity for an occupancy array is extremely important. Most obviously the granularity places a lower bound on the error since, as already mentioned single non-orthogonal transformations are inaccurate to within one element of the array. The choice of granularity also determines the increment by which component occupancy arrays should be translated and rotated. The reason for this is that the motion inference procedure which implements the single occupancy constraint is dependent on objects overlapping by no more than one boundary element. This also ensures that the regions of objects cannot pass over otherwise occupied elements in the course of a single transformation increment. An appropriate value for a rotational object's angular increment is some fraction of $\frac{grainsize}{dimension}$.

## 4 Computational Efficiency and Focusing

The strongest criticism of occupancy array representations of space is the inefficiency of the representation. This is not necessarily in terms of space required to represent objects, for we have shown that in kinematic reasoning only the occupancy at the boundaries of object need be represented. Rather, it is the fact that crudely implemented incremental rotations of an object in a world occupancy array of a dimension $D$ requires $D^2$ inverse transformations per rotational increment. Although such transformations are fully parallelizable, sequential implementation of occupancy array kinematics must overcome this inherent inefficiency.

We have implemented the refined version of occupancy array and normal calculation algorithms described in the previous sections in a prototype application *KAP* (Kinematic Analysis Program). The 2-dimensional component models are constructed us-

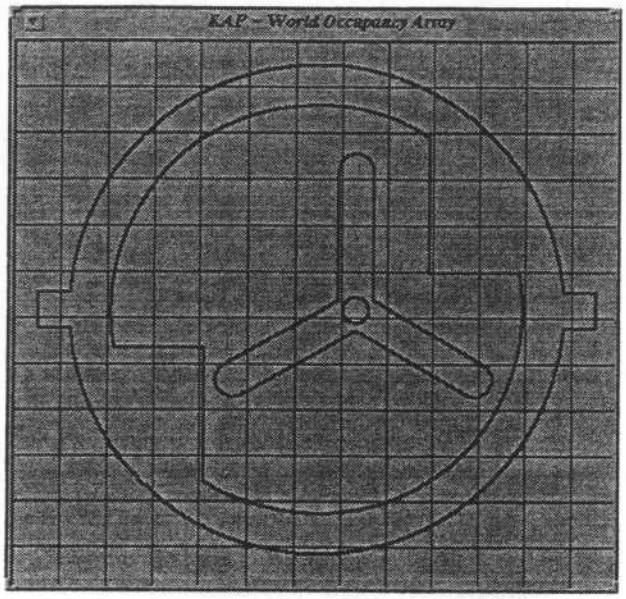


Figure 5: The world occupancy array in *KAP* for the cam-follower.

ing AutoCAD and are exported to *KAP* as bitmaps. *KAP* displays the incremental results of the occupancy array kinematic reasoning in graphical and textual form. We have tested *KAP* on a number of mechanisms including the interacting 3 gear set discussed in Section 3.4 (see figure 3). The initial lag time between gear interactions can clearly be seen in figure 4 in which the rotations generated for $C_2$ and $C_3$ are plotted against the rotation of the driving gear $C_1$.

To avoid the computational overhead in having to transform every array element of $W$ for each increment of a component's position, *KAP* uses a coarse grained occupancy array in conjunction with $W$ to dynamically focus on the regions where interactions are likely to occur.

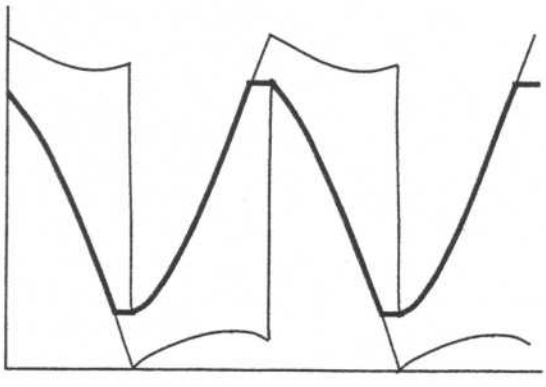

Figure 6: Configuration space for the cam-follower pair. The horizontal axis represents the rotation of the cam and the vertical axis indicates the transformational displacement of the follower.

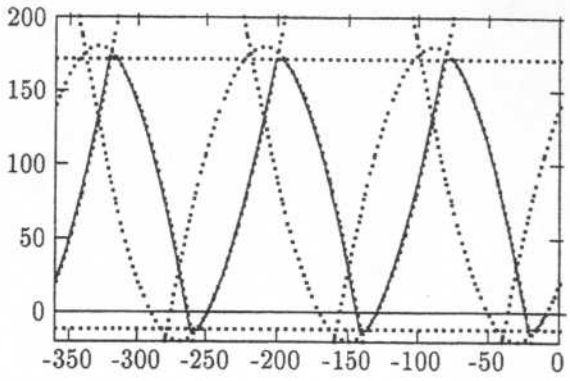


Figure 7: Simulation results from *KAP* for the cam-follower pair. The vertical axis shows the rotaton of the cam in degrees and the horizontal axis the transformational displacement of the follower in occupancy array elements. The dotted lines comprise the geometrical solution.

### 4.1 Static Focus

Focus point are identified:

1. When $W$ is initialised to contain the component occupancy arrays, construct $C$, a coarse copy of $W$. The resolution of $C$ is less than that of $W$ by some factor $f$. As in $W$, $C$ distinguishes between cells occupied by different objects, but unlike $W$, in $C$ cells may be occupied by more than one object (if in the world occupancy array the objects are proximal or touching).

2. Before any object increment, $C$ is scanned in order to identify regions in the $W$ where interactions between the components are likely. This is achieved by first creating a list $L_{focus}$ of elements of interest:

   (a) First add to $L_{focus}$ all the cells in $C$ occupied by more than one object. These are referred to as *hot spots*. For each *hot spot* add its eight neighbouring elements.

   (b) Next add to $L_{focus}$ the elements of $C$ occupied by one component which lie adjacent to cells occupied by another component.

3. Construct a minimal set of rectangular bounding boxes around the set of elements $L_{focus}$, where adjacent elements are included in the same bounding box. Each of these rectangular bounding boxes is now a focal point. Thus the temporary array $T$ of the occupancy array algorithm of Section 3.1 is no longer dimensionally identical to $W$, but instead is a collection of the bounding box regions and only in these regions are the reverse transformations performed on the array elements.

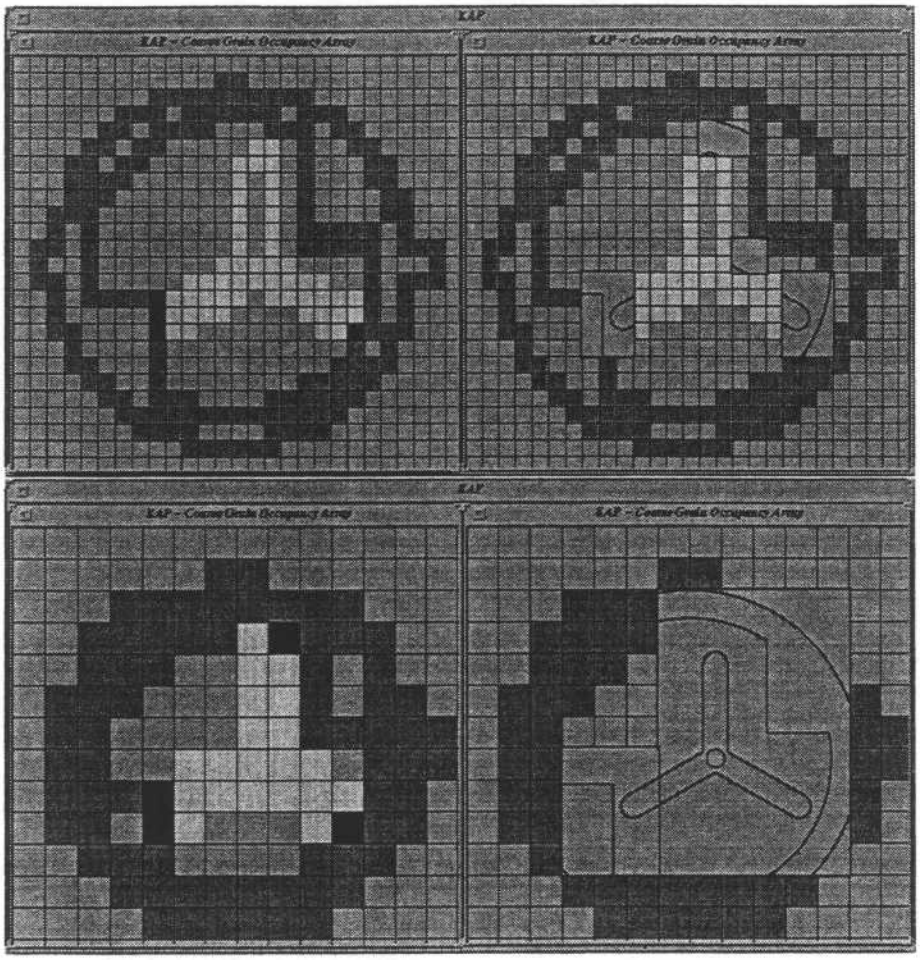

Figure 8: Images of $C$ for grain sizes 20 (top left) and 40 (bottom left). The darkest regions indicate overlap. The right hand images show the superimposed focal points. The dimension of $W$ is 600. For the top right image, the focal points comprise 40 elements of $C$, equivalent to 16000 elements of $W$, as compared to 360,000 in $W$, a saving of 95%.

The purpose of the set of focal points is to identify small areas of the $W$ that are likely to be significant, in that two different objects occupy the same region of space, and thus have the potential to interact. We illustrate focusing with the cam-follower mechanism (see figure 5, previously used by Joskowicz [1991] to exemplify the utility of the configuration space analysis of higher pairs). As the central cam rotates clockwise the follower oscillates left and right. Figure 6 shows the configuration space for the cam-follower mechanism and figure 4 is a plot of the displacement from $KAP$.

The ratio between the grain size of $C$ and $W$, and the actual spatial proximity of components in $W$ govern the efficiency improvement due to focusing. The larger the grain size of $C$ the more inclusive it is likely to be and thus less efficient due to the extended spread of the focal points. Figure 8 contrasts the different focal points for the cam-follower pair in the cases where $f = 20$ and $f = 40$.

### 4.2 Dynamic Focus

It is in the very nature of kinematic devices that their configuration changes and with it the regions likely to be the scene of the next interaction. $KAP$ accomodates this by dynamically updating the focal points as the components move. In addition, to improve the efficiency of the refocussing procedure KAP uses multiple levels occupancy arrays of gradually decreasing granularity. The intitial focus focus is constructed at the most coarse level, thereby restricting the region of search for the focus at the next level (see figure 10). How often this update is carried out is dependent in particular on the total number of motion increments for components (that is, the spatial dispersion that has occurred since the last re-focus). Because dynamic focusing relies on the spatial occupancy of a transformed $C$, larger grain coarse images tend to generate rather spurious focal points as the object displacements stray from orthogonal directions — these filtered out at the finer grained levels.

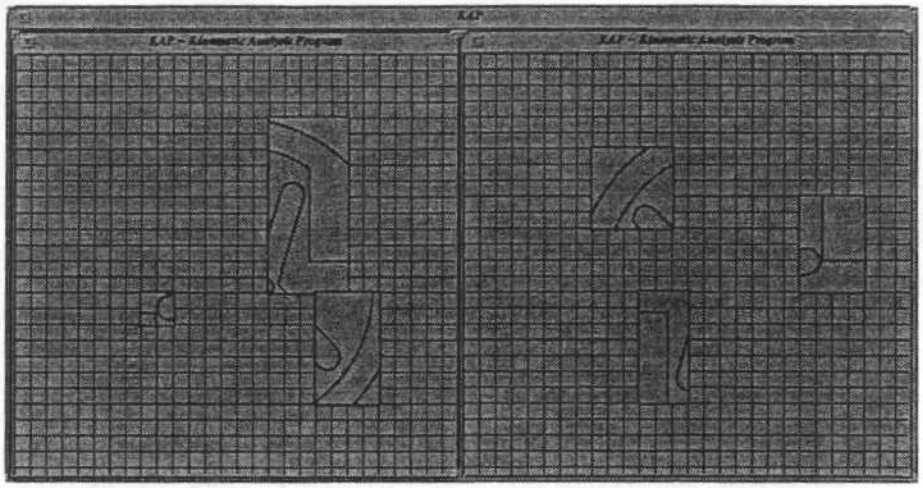

Figure 9: Two configurations for $W$ resulting from dynamic focusing.

Figure 9 illustrates two additional configurations for the cam-follower and the associated focal points dynamically created during simulation by *KAP*.

## 5 Related Work

Established work in kinematics and qualitative kinematics concentrates almost exclusively on the configuration space (CS) approach [Forbus *et al.*, 1987, Faltings, 1990,
Faltings, 1992, Joskowicz and Sacks, 1991]. Yet occupancy arrays share little in common with configuration spaces which are based on the solution of equation sets describing the object boundaries for all possible configurations. Computing the CS for a mechanism does however rely on having an accurate analytical description of the object (although linear approximations can be compounded to capture complex features). With respect to this point the occupancy array approach has a far higher aquisitional efficiency.

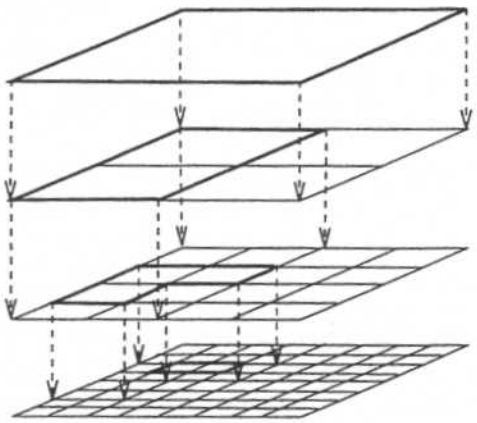

Figure 10: Efficient refocusing through multiple levels of granularity.

Of previous work in the field of spatial reasoning, ours is closest to Funt's WHISPER system [Funt, 1976, Funt, 1980] which indeed provided some of our initial motivation. Funt used very coarse occupancy arrays to predict motion induced in stacked blocks by instabilities. Narayanan [1991, 1992] uses diagrams to represent visual information utilised by the inference rules which provide knowledge-level representation of physical interactions. Our approach does not use inference rules but works simply on the constraint that no two objects occupy the same position in the occupancy array. This same approach is taken by work in [Gardin and Meltzer, 1989] who realise a naive physics of the behaviour of flexible objects and their interaction with rigid objects.

More recently, Glasgow's [1993, 1994] symbolic arrays are even more descriptive still, in fact a symbol in an array can often represent an entity by itself; this is sufficient for reasoning about gross positional relations, but unsuitable for physical interactions where knowledge of shape is crucial. Glasgow's three level spatial model does include the occupancy array at the lowest level, though she is non-commital as to its role in reasoning.

## 6 Conclusion and Future Work

We have shown that by representing the spatial occupancy of objects using appropriately high resolution occupancy arrays we can reason about higher pair kinematic interactions between objects. Both free and blocked motion can be inferred and we report an implementation, *KAP*, that demonstrates the utility of this approach.

Inferring the transmitted motion as a result of an interaction between occupancy arrays in principle depends on the granularity of the array with respect to the radius of curvature of boundary features, and in the case of objects with rotational degrees of freedom, the distance of the boundary features from the axis of rotation. We plan to conduct an analysis of the

sensitivity of the algorithm with respect to this.

We also intend to enhance the robustness of the local search procedure with respect to discontinuities in the object outline (such as sharp corners). Long term goals include the extension of *KAP* to 3D and modelling linkages.

## References


[Davis, 1990] Ernest Davis. *Representations of Commonsense Knowledge.* Morgan Kaufmann, 1990.

[Faltings, 1990] Boi Faltings. Qualitative kinematics in mechanisms. *Artificial Intelligence*, 44:89–120, 1990.

[Faltings, 1992] Boi Faltings. A symbolic approach to qualitative kinematics. *Artificial Intelligence*, 56:139–170, 1992.

[Forbus *et al.*, 1987] Kenneth D Forbus, Paul Neilsen, and Boi Faltings. Qualitative kinematics: A framework. In *Proceedings Tenth International Joint Conference on Artificial Intelligence*, pages 430–436, Milan, Italy, 1987.

[Funt, 1976] Brian V Funt. *WHISPER: A Computer Implementation Using Analogues in Reasoning.* PhD thesis, University of British Columbia, 1976.

[Funt, 1980] Brian V Funt. Problem-solving with diagrammatic representations. *Artificial Intelligence*, 13:201–230, 1980.

[Gardin and Meltzer, 1989] Francesco Gardin and Bernard Meltzer. Analogical representations of naive physics. *Artificial Intelligence*, 38:139–159, 1989.

[Glasgow, 1993] Janice I Glasgow. The imagery debate revisited. *Computational Intelligence*, 9(4):309–333, November 1993.

[Glasgow, 1994] Janice I Glasgow. Spatial reasoning in indeterminate worlds. In *Proceedings Twelfth National Conference on Artificial Intelligence*, pages 1399–1404, Seattle, 1994. AAAI/MIT Press.

[Joskowicz and Sacks, 1991] Leo Joskowicz and Elisha Sacks. Computational kinematics. *Artificial Intelligence*, 51:381–416, 1991.

[Narayanan and Chandrasekaran, 1991]
N Hari Narayanan and B Chandrasekaran. Reasoning visually about spatial interactions. In *Proceedings Twelfth International Joint Conference on Artificial Intelligence*, Sydney, Australia, 1991.

[Narayanan, 1992] N Hari Narayanan. *Imagery, Diagrams and Reasoning.* PhD thesis, Ohio State University, 1992.