Parallel Image Matching on Distributed Shared Memory Network

J. You[1] W.P. Zhu[2] H.A. Cohen[3]
[1]*School of Computer and Information Technology*
Griffith University, Queensland, Australia 4001
E-mail: you@kurango.cit.gu.edu.au
[2]*School of Computer and Information Science*
University of South Australia, The Levels, South Australia, 5095
E-mail: cisuipz@cis0.levels.unisa.edu.au
[3]*Dept. of Computer Science and Computer Engineering*
La Trobe University, Bundoora, Victoria, Australia 3083
E-mail: harvey@latcs1.lat.oz.au

Abstract The need to implement image processing tasks in parallel arises from the speed requirements of real-time environments in various application domains. In contrast to the traditional parallel solutions which relied on specialized multiprocessor architectures, we propose a hierarchical image matching scheme for effective and efficient object recognition by using the advanced technology of computer network as an appealing tool for parallel computing at a relatively low cost. This paper presents our investigation of parallel implementation on networks of workstations using distributed shared memory (DSM). The challenge of a DSM system is that the shared memory abstraction does not result in a large amount of communication by applying lazy release consistency and multiple writer protocols. Furthermore, we extend our previous work by introducing Hausdorff distance to handle occluded objects. The hierarchical matching is guided by the detection of interesting points. A divide-and-conquer policy is adopted in such a scheme, where a complex task is divided into a number of sub-tasks and those sub-tasks are later reorganized into clusters according to granularity before being mapped on computers for simultaneous implementation. The system performance is evaluated in terms of recognition accuracy and execution time. We conclude that distributed shared memory computing can meet the high computational and memory access demands in real-time imaging.

Keywords and phrases: object recognition, image matching, feature extraction, parallel processing, distributed shared memory.

1 Introduction

Image matching is a key problem in pattern recognition, image analysis and computer vision. The current goal of research in the field of image matching is to find an efficient approach to measure the degree of similarity between two image sets that are superimposed upon one another. In general, matching two images requires that the images to be matched go through a number of stages of processing including the image feature extraction, matching measurement and searching for the best match. In the past many sophisticated matching algorithms have been proposed, such as Chamfer matching[1], Borgefors matching scheme[2] and the Hausdorff distance[3]. In each of these schemes edge points are detected as image feature points and a distance transform is applied to these binary images so that certain matching criteria can be applied to measure the similarity between the two image sets. However, the applications of these methods are restricted due to the following disadvantages:

- redundant or mis-detected feature points (from the use of edge detection)
- failure to deal with non-rigid objects, *i.e.* objects which are not clearly defined (Borgefors matching scheme)
- failure to deal with partial, or partially hidden objects (Chamfer matching, Borgefors scheme)
- time-consuming for point-wise shape comparison (Hausdorff distance)

While image matching has many potential applications, it is computationally costly to have the matching algorithm determine a value as to how good the match is for every possible position combination of the template window within the target image. A better solution would be to have a guided search where the likely matching areas are selected and values determining how good a match is are only computed for the likely match areas. This is the concept behind a hierarchical image matching. However, it is too slow to be useful within any environment approaching real-time requirements.

This paper presents an efficient and effective image matching scheme for object recognition. Based-on our previous work[4], we aim to develop a guided object recognition system by implementing a hierarchical image matching scheme in parallel on a distributed shared memory network. Instead of using message passing as the parallel programming paradigm for a workstation cluster, we investigate parallel computing on networks of workstations using distributed shared memory (DSM). The challenge of a DSM system is that it reduces communication overheads significantly to speed up the execution by applying lazy release consistency and multiple writer protocols.

In the context of this paper, Section 2 summarises the proposed hierarchical image matching scheme and Section 3 highlights the parallel implementation using distributed shared memory workstations. Finally, our experimental results and conclusions are presented in Section 4 and Section 5 respectively.

2 Hierarchical Image Matching

The hierarchical image matching scheme was first proposed by Borgefors[2] in order to reduce the computation required to match two images. This Section details our extension of this scheme which uses the Hausdorff distance rather than Chamfer matching as a measure of similarity between the images. In addition, a guided search strategy is introduced to avoid the blind searching for the best fit between the given patterns.

2.1 Matching Measurement — The Hausdorff Distance

The Hausdorff distance is a non-linear operator which measures the mismatch of the two sets. In other words, such a distance determines the degree of the mismatch between a model and an object by measuring the distance of the point of a model that is farthest from any point of an object and vice versa. Therefore, it can be used for object recognition by comparing two images which are superimposed on one another. The concept of such measurement can be summarised as below.

Given two finite point sets $A = \{a_1, ..., a_m\}$ and $B = \{b_1,.....b,\}$, the Hausdorff distance D_H between these two sets is defined as

$$D_H = \max(d_{AB}, d_{BA})$$

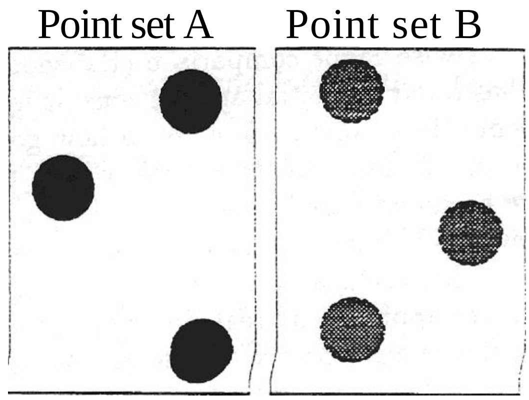
where d_As is the distance from set A to set B expressed as

$$d_{AB} = \max_{ai \in A}(d_{aiB})$$

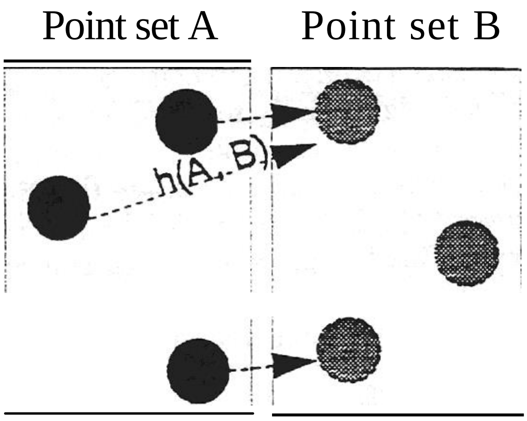
while $d_{a_i B}$ is the distance from point a_i to set B given by

$$d_{a_i B} = \min_{b_j \in B}(d_{a_i b_j})$$

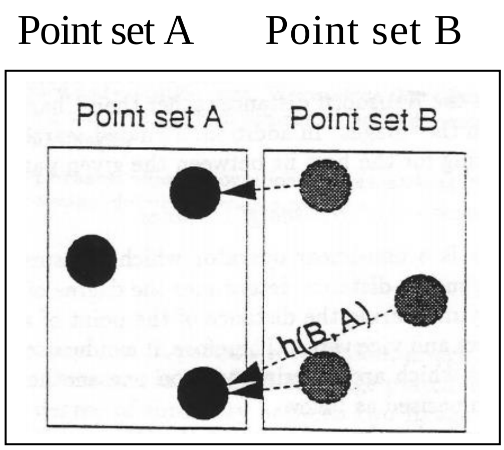
Obviously the Hausdorff distance D_H is the maximum of d_{AB} and d_{BA} which measures the degree of mismatch between two sets A and B. Figure 1 presents a graphical example of computing the Hausdorff distance.



the two point sets A and B



the directed Hausdorff distance h(A, B)



the directed Hausdorff distance h(B, A)

Figure 1 A graphical example of computing the Hausdorff distance
note *H (A, B) = max(h(A, B), h(B, A))*

2.2 The Computation of Hausdorff Distance

For a feature points detected image, the characteristic function of the set A and B can be represented by a binary array A[i,j] and B[i,j] respectively, where the (i, j)th entry in the array is non-zero for the corresponding feature points in the given image. Therefore, distance array D[i,j], D'[i,j] should be obtained to specify for each pixel location (i,j) the distance to the nearest non-zero pixel of A or B respectively, where D[i,j] denotes the distance transform of A and D'[i,j] denotes the distance transform of B. Consequently, the Hausdorff distance as a function of translation can be determined by computing the pointwise maximum of all the translated D and D' array in the form of:

$$F[i,j] = \max(\max_a, \max_b)$$

where

$$\max_a = \max_a D[a_i - i, a_j - j]$$
$$\max_b = \max_b D[b_i + i, b_j + j]$$

2.3 An Improved Matching Scheme

The Hausdorff distance is viewed as a better solution to image matching because it is relatively insensitive to small perturbations in images and natural allowance for portions of the image to be compared[3]. However, the processing is very time-consuming as more edge points are involved in distance transform for the purpose of accuracy. In order to reduce the number of feature points in image without losing information for distance transform, we introduce a dynamic procedure to detect interesting points as image feature points to determine the Hausdorff distance for matching. The detection of interesting points is based on the measure of how interesting a point is and an interesting point should be regarded as distinctness, invariance, stability, uniqueness and interpretability. In general, the detection of interesting points can be summarised as a three-step procedure:

- Selection of optimal windows. The selection is based on the average gradient magnitude within a window of prespecified size. Search for local maxima, while suppressing windows on edges and guaranteeing local distinctness. The measure used should also be invariant of rotation.
- Classification of the image function within the selected windows. The classification distinguishes between types of singular points such as corners, rings, spirals, and even isotropic texture based on a statistical test.
- Estimation of the optimal point within the window as the classification. The estimation is precise for corners and for the centers of circular symmetric features or spirals.

In contrast to the traditional concept of edge distance for matching measurement, we replace edge points with interesting points to calculate the Hausdorff distance as the matching measurement, which improves the efficiency and reliability of the testing results. Figure 2 shows the comparison of edge points and interesting points in representing the original image. Figure 2(a) is a histogram equalized image of size 256 x 256 ranging from 0 - 255 in gray scale, Figure 2(b) shows edge points detected by Prewitt operator, Figure 2(c) shows interesting points detected by Moravec operator, and Figure 2(d) shows interesting points detected by Plessey operator. It is clear that the complexity of computation for matching depends on the image feature pixels and interesting points detected by Plessey best reserve the image features with less points.

2.4 A Guided Search Procedure

In order to avoid the blind searching for the best fit between the given patterns, a guided search strategy is essential to reduce computation burden. Our extension of the hierarchical image matching scheme (H.I.M.S) was based on a guided searching algorithm

(a) Original image (256 x 256)   (b) Edge points (Prewitt operator)

© Interesting points (Moravec operator)   (d) Interesting points (Plessey operators)

Figure 2. Comparison of image feature extraction operators

that searches first at the low level, coarse grained images, to the high level, fine grained images. To do this we needed to obtain a Hausdorff distance approximation for each possible window combination of the template and target image at the lowest resolution. Those that returned a Hausdorff distance approximation equal to the lowest Hausdorff distance for those images were investigated at the higher resolution. The following summarises the key steps involved in a H.I.M.S algorithm, while Figure 3 shows a binary image pyramid for guided search.

```
create image pyramid
     for all combinations of windows at lowest level
     get value of match for this combination
    if low value add to lowest list
end-for

for each remaining level
    remove area from lowest list
    get match value for this area
    if low value add to lowest list
end-for
```

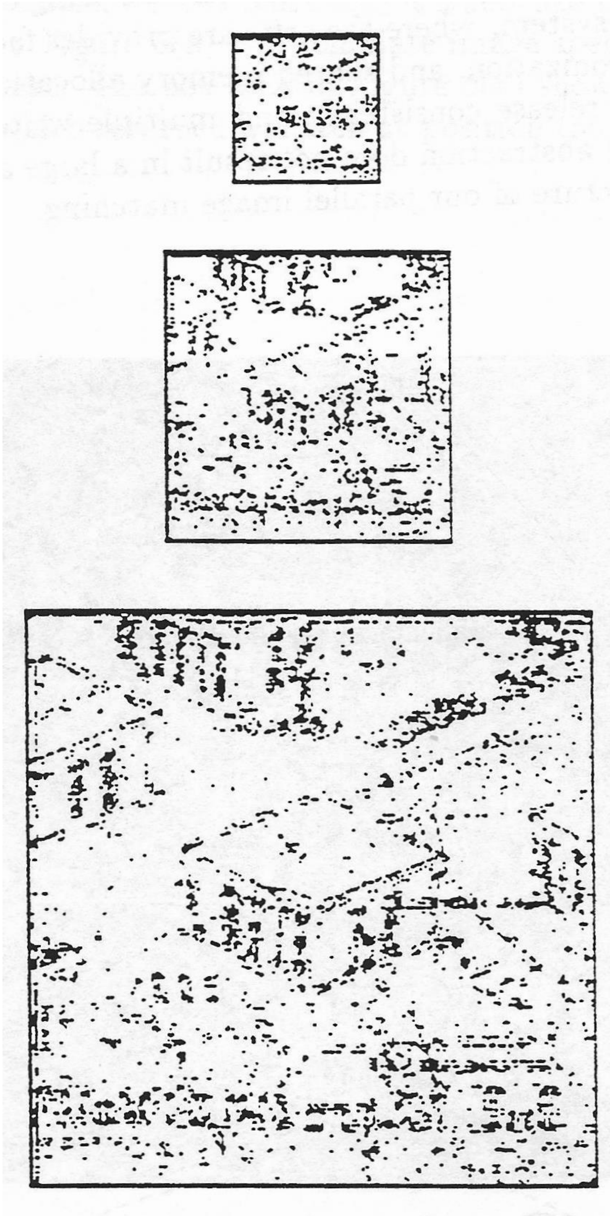


Figure 3. A binary image pyramid

3 Parallel Implementation

Parallel computation has been used successfully in many areas of computer science to speed up the computation required to solve a problem. In the field of image processing and computer vision this is especially appropriate since it appears that the biological model for vision is a parallel model.

In an attempt to meet the real time requirement and decrease execution time the matching scheme detailed in this paper can be implemented in parallel using a workstation cluster. In our previous work, we applied PVM(Parallel Virtual Machine) to support message passing for distributed memory systems. In order to further reduce the communication overheads, we extend our experiment of parallel computing on networks of workstations using the TreadMarks distributed shared memory (DSM) system{5].

TreadMarks is a useful software which turns an existing network of workstations into a powerful shared-memory parallel computer for high performance. Based on the workstation's unmodified Unix operating system and standard compilers, TreadMarks creates a parallel programming environment that makes parallel computing convenient, inexpensive and efficient. In this section, we discuss our experience with parallel implementation of hierarchical image matching on a network of workstations using the TreadMarks distributed shared memory (DSM) system. where the software provides facilities for process creation and destruction, synchronization, and shared memory allocation. The TreadMarks DSM system introduces lazy release consistency and multiple writer protocols to make sure that the shared memory abstraction does not result in a large amount of communication. Figure 4 shows the structure of our parallel image matching.

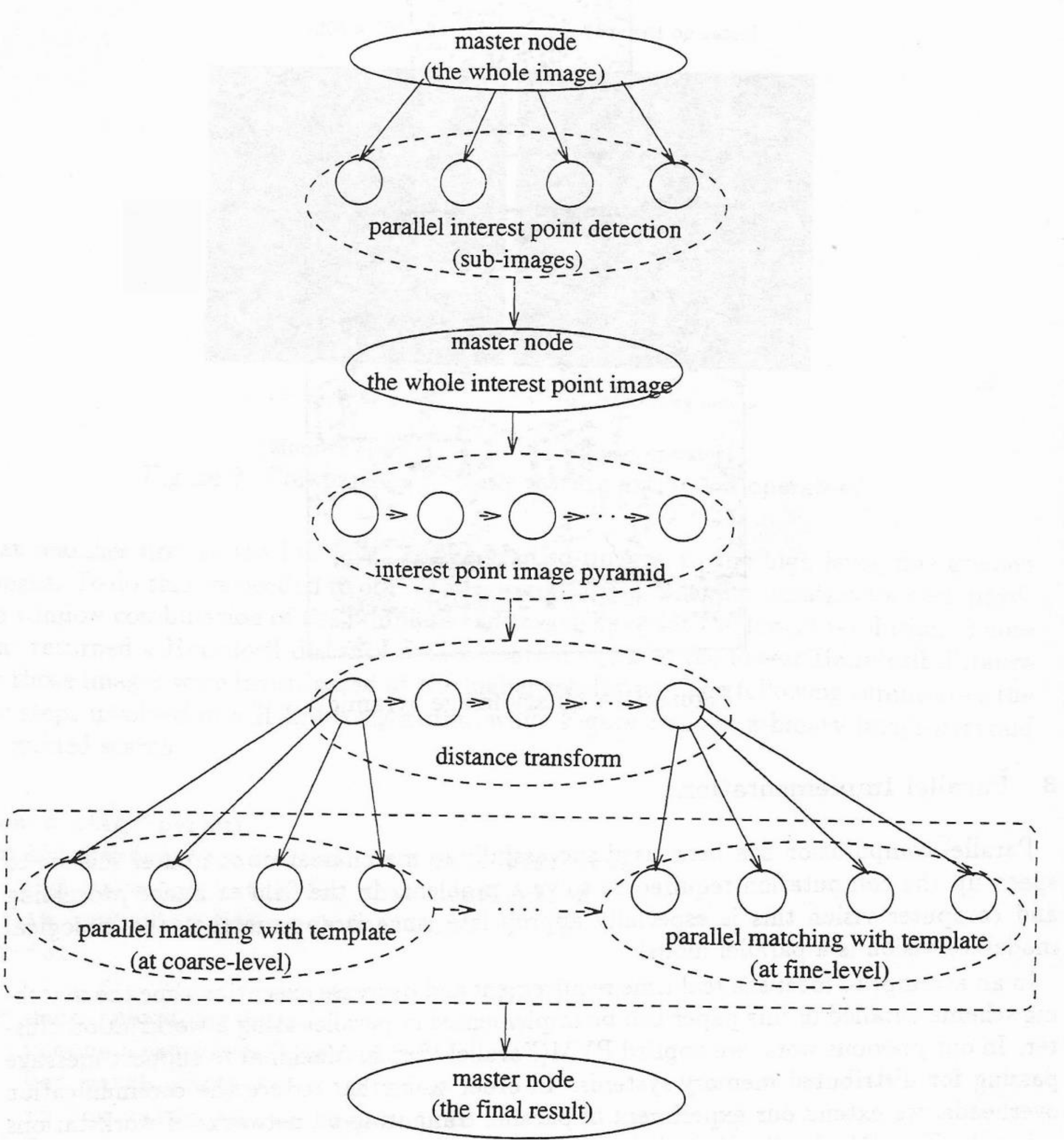
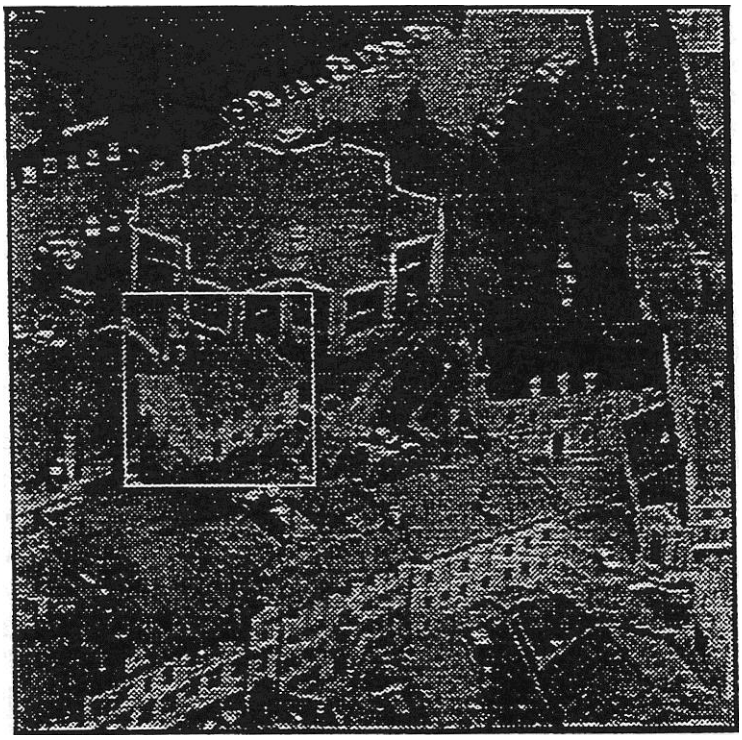


Figure 4. The system structure for parallel implementation

4 Experimental Results

Our hierarchical matching scheme can be applied to object recognition and localisation. Figure 5 shows such an example, where Figure 5(a) is a 300320 target image with certain object to be identified and Figure 5(b) is a template image specifying the objects to be expected in the target image. The box area in Figure 5(a) shows the matching result of our hierarchical scheme, which returned a match at position (56, 142).



(a) Target image

(b) Template image

Figure 5. Object recognition and localisation by matching

The advantages and potentials of a general distributed shared memory system for parallel image processing is demonstrated by the performance comparison listed in Table 1, where different numbers of processes were invoked by means of both PVM and DSM on the 512*512 size image. The parallel performance improvement is measured by the following ratio:

$$\gamma = T_P / T_S$$

where T_S refers to the sequential execution time and T_P refers to the parallel execution time. It is clear that DSM increase the speed better due to its less communication overheads. It is expected that the overall performance will be further improved by introducing dynamic task scheduling for parallelism.

Table 1: The performance comparison of DSM and PVM

number of processes	execution time (DSM)	execution time (PVM)
1	1.45 sec.	1.77 sec.
2	0.80 sec.	1.41 sec.
4	0.52 sec.	1.17 sec.
8	0.48 sec.	1.13 sec.
16	0.22 sec.	0.27 sec.
32	0.12 sec.	0.27 sec.

5 Conclusion

We conclude that interesting points are better feature extractors than edge detectors because they return fewer points which speeds up the matching scheme. We can also conclude that the Hausdorff distance is a quick and accurate method of determining the degree of similarity between two images. By combining these two approaches within a hierarchical image matching scheme we obtained a great improvement in the overall computation time. Furthermore, the parallel computing on networks of workstations using distributed shared memory provides a further improvement in the computation time to attain real-time performance. Our investigation shows that distributed shared memory computing can meet the high computational and memory access demands in real-time imaging.

References

[1] H.G. Barrow, J.M. Tenenbaum, R.C. Bolles and H.C. Wolf, "Parametric correspondence and chamfer matching: Two new techniques for image matching", *Proc. 5th Int. Joint Conf. Artificial Intelligence,* Cambridge, MA, pp. 659-663, 1977.

[2] G. Borgefors, "Hierarchical chamfer matching: a parametric edge matching algorithm", *IEEE Trans. Patt. Anal. Machine Intell.,* Vol. PAMI-10, pp. 849-865, 1988.

[3] D.P. Huttenlocher, G.A. Klanderman and W.J. Rucklidge, "Comparing images using the Hausdorff distance", *IEEE Trans. Patt. Anal. Machine Intell.,* Vol. PAMI-15, pp. 850-863, 1993.

[4] J. You, W.P. Zhu, E. Pissaloux, and H.A. Cohen, "Parallel image matching on a distributed system", *Proc. of IEEE First International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP-95),* Vol. 2, Brisbane, Australia, April 19-21, 1995, pp. 870-873.

[5] Willy Zwaenepoel *et.* al., "TreadMarks: Shared memory computing on networks of workstations", *Technical Report,* Dept. of Computer Science, Rice University, 1994.

[6] H.P. Moravec, "Towards automatic visual obstacle avoidance", *Proc. 5th Int. Joint Conf. Artificial Intelligence,* Cambridge, MA, pp. 584, 1977.

[7] A. Rosenfeld, "Multiresolution image representation", in *Digital Image Analysis,* S. Levialdi, Ed., London: Pitman, pp. 18-28, 1984.

[8] R.M. Haralick and L.G. Shapiro, *Computer and Robot Vision,* Vol. 2, Addison-Wesley Publishing Co., 1993.