Ref: "Barnard Thompson. to obtain sparse disparity map" Monvee.

Implementation of a Feature Point Stereo Image Matching Algorithm on a Transputer Network*

by

K.A. Collins** , J.B.G. Roberts, K.J. Palmer

Royal Signals and Radar Establishment,
St. Andrews Road,
Great Malvern,
Worcestershire,
U.K.,
WR14 6BT.

ABSTRACT

This paper will focus on the issues concerned with a parallel implementation strategy for the feature point stereo image matching algorithm published by Barnard and Thompson (See Barnard and Thompson 1981). The performance and software engineering aspects of the implementation on a parallel computer are discussed. This implementation is expected to provide a foundation for one processing stage in the development of a Transputer-based system for real-time production of depth maps from a variety of data sources.

The implementation strategy utilises the asynchronous SIMD mode of processing in which the same program is executed on each processor, but not in "lock-step". A simple framework has been devised that supports all necessary modes of interprocessor communication, and permits parameterisation of the logical and physcal configuration of the array. Preliminary results on speed will be given, and their effect on future implementation strategies discussed.

I INTRODUCTION

In the real world, sensor geometry cannot be assumed to be epipolar e.g. SPOT (See Chevrel et al 1981, Dowman 1982). Barnard and Thompson's algorithm is capable of processing stereo imagery captured under such conditions, and generates a sparse disparity map which may be used to refine estimates for the relative orientation parameters of the sensor. The typical processing sequence in order to produce epipolar SPOT stereo pairs would be:

1. Input raw SPOT stereo pairs and platform parameters.
2. Calculate rough estimates of absolute orientation parameters.
3. Define overlapping areas of interest in stereo pair.
4. Apply Barnard and Thompson to rough epipolar SPOT stereo pair, to obtain a sparse disparity map.
5. Refine sparse disparity map to required accuracy for step 6 and 7.
6. Calculate resampling transformation required to produce epipolar SPOT stereo pairs.
7. Resample to epipolar.

An adequate specification of the algorithm existed in the literature, permitting direct implementation in Occam2, the target machine being a Meiko Computing Surface. This approach permitted the competence of the original algorithm and the speed of the parallel implementation to be assessed together. This is a useful approach: quality and speed metrics are dependent upon each other, and algorithm and implementation issues can be studied together rather than in isolation. Following this evaluation, recommmendations to improve both the competences and speed of the algorithm are to be made based upon the results to date.

* This work is funded by DTI under the Alvey MMI -137 project.

** Seconded to RSRE from THORN EMI Central Research Laboratories, Dawley Road, Hayes, Middlesex, U.K. UB3 1HH.

AVC 1987 doi:10.5244/C.1.21

II DEVELOPMENT ENVIRONMENT

Initial development of a sequential implementation was carried out using the INMOS Transputer Development System hosted on an IBM PC AT / INMOS B004 board, using an INMOS B007 graphics board and colour monitor for displaying results. Subsequent development of a parallel implementation made use of the aforementioned host and display system plus a Meiko Computing Surface consisting of:

i) Meiko host board (1 Transputer with 3Mbytes DRAM) and dumb terminal,

ii) 5 Meiko compute boards (total of 20 "worker" Transputers, each with 256K bytes DRAM).

15 MHz T414 (Rev B) Transputer parts have been used here. Therefore, link bandwidth is 10Mbits/s and all floating point calculation is done in software. The advent of the T800 Transputer will double link bandwidth and provide an increase of 6-10 times the speed in floating point performance. Both these aspects of the T800 performance should reduce processing time significantly.

III ALGORITHM SUMMARY

The algorithm implemented is as specified by Barnard and Thompson. The algorithm consists of four sequential phases:

Phase 1. Feature Extraction using the Moravec Interest Operator (See Moravec 1977). The Moravec operator detects unique feature points in an image based on the assumption that a pixel is interesting if it has a high variance in all directions. The sum of absolute values of greylevel differences in the four principle directions are calculated, and the local minimum taken as the value for that pixel. Subsequently these values are searched for local maxima that exceed a pre-defined threshold, using an overlapping suppression window, which are then taken as the unique feature points. This operator is applied to both the reference (left) and match (right) images in order to extract the feature points required for the matching phases.

Phase 2. Global Match Network Construction. A global match network is constructed that contains all candidate matches. For each reference feature point a local match sub-network is generated, which contains all candidate matches for that feature. Each candidate match is represented by the pairing of a feature from the reference image with a feature point from the match image, or with an 'undefined' label indicating that the feature is unmatchable. For each candidate match an initial match probability is generated. For features that are well-defined, this estimate is based on a similarity measure calculated using the sum of squares of differences of pixels in windows of raw image data centred over each pair of corresponding feature points. For the undefined label, this probability is generated based upon the best similarity measure observed for a well-defined candidate match in the local match sub-network. The number of candidate matches generated is dependent upon the maximum permissible horizontal (x) and vertical (y) disparity, and the spatial density and distribution of features from phase 1. The disparity limits are set a priori to constrain the size of the disparity space to be searched.

Phase 3. Global Match Network Update. The most likely match for each reference image feature point is determined through the iterative application of a relaxation operator in the local neighborhood of the reference feature. The consistency of disparities is used as the condition to update match probabilities by searching all candidate matches in the neighborhood of the candidate match to be updated, and finding those for which the disparity difference is less than some threshold. This surface consistency threshold and the size of the neighborhood used dictate the shape of the surface (in disparity space) that is permitted by the algorithm. The x,y dimensions of the neighborhood used and the spatial distribution and density of the candidate matches resulting from phase 2, dictate the number of candidate matches examined during this phase. The surface consistency threshold determines how many of the matches examined contribute to the new probability estimate. The number of iterations of the relaxation operator is a predefined constant.

Phase 4. Selection of Correspondences by Thresholding. The probability of the most likely correspondence in each match sub-network is thresholded to obtain a list of unique correspondences.

IV ALGORITHMIC DEFICIENCIES

The most important characteristics of an interest operator for detecting feature points are that the points chosen should be distinct and appear in both images, and the position should be invariant to geometric and radiometric distortions. When applied to simulated SPOT data, the Moravec operator frequently does not produce consistent or invariant features in each image. At present modifications to the operator to improve its competence, and alternative interest operators (See Forstner 1986), are being examined.

The result of this deficiency is that no well-defined matches are produced by the algorithm on simulated SPOT data. Clearly, if a feature does not have a corresponding feature in the match image, it cannot be matched, and does not strengthen the match probabilities at neighbouring match sub-networks. If a feature does have a corresponding feature in the match image, but its selected position is not invariant, the initial match probability generated will be small. As a result, convergence to the correct disparity is unlikely to occur and the disparity will not be consistent with its neighbors.

During phase 2, a similarity measure is used to generate initial match probabilities. This measure is calculated under the assumption that there are no geometric or radiometric distortions present. This is unlikely to be the case in practice where images may have been captured days apart and parameters of the sensor geometry are only known approximately. The effects of perspective distortion are likely to be small due to the small window size used, but radiometric distortion is likely to be significant factor affecting if, and how quickly, the relaxation stage will converge.

During phase 3, for a local neighborhood of fixed size, and if the feature detector provides sufficient consistent and invariant features, then the density of reference feature points and the disparity surface model used will dictate how rapidly the relaxation stage will converge. In general, it is expected that if there is a high density of consistent matches the match network will approach the steady-state condition quickly. It is also expected that there is a minimum or "critical" density of points required in order for convergence to occur.

The disparity surface model suggested by Barnard and Thompson is extremely simple. The condition used is that disparity differences of 1 pixel within a neighbourhood of r pixels (r = 12 pixels) is considered consistent. A better idea is to use a 'disparity gradient limit' approach (See Pollard et al 1985). Candidate matches would be considered consistent if the disparity gradient between them falls below a certain threshold. The contribution of consistent matches to the update might be weighted as a function of the inter-point distance and the observed disparity gradient. In this way, the effective size of the neighborhood used during phase 3 might be increased, thus reducing the density of points required for convergence.

V THE APPLICATION OF PARALLEL PROCESSING TO STEREO MATCHING ALGORITHMS

It is clear that the computational complexity of stereo algorithms, and in particular the preprocessing and disambiguation phases, warrant the application of parallel processing techniques in order to meet the speed requirements of a "real-time" system. From a computational viewpoint stereo algorithms are of interest due to the data dependent processing required by different scenes and camera geometries. This is particularly apparent in the case of Barnard and Thompson's algorithm where phases 2 and 3 (Global Match Network Construction and Update) require image dependent access to data. This is in contrast to phase 1 (Feature Extraction), which like many low level image processing tasks requires access to data in a predictable, data dependent manner, and the processing time per pixel is roughly constant. (See Harp et al 1985) In this case, only the suppression operation is data dependent, a pixel is processed if it is above the designated threshold. Phases 2 and 3 require access to data in a data dependent manner, but the search space for data is limited by parameters of the algorithm i.e. disparity limits, update neighborhood dimensions. This has permitted a similar implementation strategy to be applied to phases 2 and 3 as is conventionally applied to phase 1: the asynchronous SIMD strategy.

VI IMPLEMENTATION SUMMARY

The chosen strategy divides each image into horizontal strips, each strip consisting of several rasters. The processors are configured in a linear chain. Each is allocated the corresponding strips for the image pair, such that the first in the chain receives the top strip of each image, and so on for each processor in the chain. This strategy maintains the position of pixels within the images relative to the position of the processor within the chain. This permits the required inter-processor link communication bandwidth to be minimised. In addition it is assumed that the x,y dimensions of the sub-image are greater than or equal to:

i) the maximum x and y disparity used in phase 2,

ii) the maximum x,y dimension of the update neighbourhood used in phase 3.

This assumption permits all inter-processor communication during intermediate phases of the computation to be restricted to nearest-neighbour. During the input and output phases of the computation, only a 'store and forward' type inter-processor communication is required. As a result, programming complexity is reduced, a simple process hierachy provides a modular framework that supports the two required modes of communication.

In general, this type of approach is typified by the use of geometric, or data, parallelism in which processors are directly coupled by explicit synchronisation 'pulses' during the computation. These pulses are required during phase 3 when the data structures representing the match sub-networks held by a processor's nearest-neighbours are updated at the end of a relaxation cycle. The synchronisation pulses ensure that the updates occur only when the processor is prepared to accept the data i.e. the processor has finished its current computational cycle. Clearly, the synchronisation time represents wasted computation time due to poor distribution of workload. During phase 2 and 3, the number of reference features can be used as a first order estimation of a processor's workload. As a result the density and distribution of feature points, as a function of the interest operator parameters, is important to both the operation of the algorithm and the efficiency of the implementation.

This is in contrast to another SIMD strategy known as the processor farm in which data parallelism is used, however processors do not handshake directly with each other, but do so via a global host which maintains a record of the current state of the computation. This approach is the other obvious candidate for implementing this algorithm. The farm has the advantage that the communications bandwidth of the Transputer network can be more fully utilised as no explicit synchronisation between processors is needed. The farm has the disadvantage that the total required communications bandwidth is substantially higher than the present implementation. If it is found that the communications/synchronisation cycle time during phase 3 of the computation is becoming dominant, causing excessively low processor utilisations, then the farm approach will be worth experimenting with.

VII PROCESSING SEQUENCE

The following processing sequence implements the four phases of Barnard and Thompsons algorithm:

i) Pre-buffering Phase. The host file server reads the stereo image pair from disk and transmits them, raster by raster, down the output data link leading to the Meiko host board, where the image data is buffered prior to phase 2.

ii) Input Phase. The images are transmitted, raster by raster, down the processor chain. Each "worker" processor buffers the appropriate rasters (including raw data overlap) according to its position in the chain. Rasters are displayed as they reach the graphics node.

iii) Phase 1. Once all image data has been forwarded, each processor calls the Moravec operator in parallel to process the two local sub-images concurrently. Once the local processing of each sub-image is complete i.e. a list of feature points exists, these lists are exchanged between processors in order that the non-local maxima suppression operation is completed across processor borders.

iv) Phase 2. This phase begins with local processors swapping lists of the extracted feature positions, and the appropriate 5 by 5 pixel windows centred over the features. Then a computational phase begins in which, for each locally detected feature, all possible correspondences are found by searching the local and non-local lists for features within the given disparity limits, and the initial match probability for each trial correspondence is calculated.

v) Phase 3. This phase consists of a loop, which is executed a fixed number of times, determined by the number of relaxation cycles specified a priori. Each cycle consists of swapping local match sub-networks between processors, and then using this non-local data in conjunction with the local data to update all local match sub-networks. It should be noted that, at present, all arithmetic during the update phase is in REAL32.

vi) Phase 4. For each local match sub-network, the maximum match probability is found, and thresholded. The resulting list of matches represents the output data of the processor.

VIII A FRAMEWORK FOR PARALLEL IMPLEMENTATION

As indicated above, only simple modes of communication are required. These are supported within a three layer process shell which are interfaced using Occam channels. The first, outermost process is responsible for handling all link communications, multiplexing physical links to/from an internal Occam channel. This communication kernel process is run at high priority to ensure that link input/output is given precedence. The second, middle process is responsible for managing all data, based on data tagging. Two primary tag types exist, 'external' and 'internal'. External tags specify store and forward data types. Internal data tags specify data types which are passed between processors on a nearest-neighbour basis. The data management process uses these tags to call the appropriate management routine and/or output the appropriate channel synchronisations to the main computational process. The main computational process is the third, innermost process. This process calls computational routines based upon channel input synchronisations from the data management process and produces channel output synchronisations to the same process when finished e.g. a call to the Moravec routine in pseudo-Occam,

```
SEQ
    from.data.manager ? raw.data.ready
    Moravec( raw.data, feature.list )
    to.data.manager ? feature.list.ready
```

This three layer structure permits all computational routines to be written independently of logical/physical topology, and without embedded channel/link communication statements.

IX PRELIMINARY SPEED RESULTS

In the previous discussion it is clear that the number of parameters that affect the observed speed is large: speed is dependent on the parameters of the algorithm and characteristics of the stereo pair e.g. the scene, camera geometry and illumination conditions observed. To date, the important objectives of speed assessment have been to identify program bottlenecks and generate rules of thumb to enable prediction of the behavoir of speed and efficiency metrics for the current implementation with respect to the aforementioned parameter space. Here, the important metrics presented for a given parameter set are the average processor utilisation during phase 3, and the average processsing time per point.

All timings shown below are for stereo image pairs, and are in milliseconds. For each image, the total area processed is 240 by 226 pixels. Both sets of timings given here are for the full implementation of Barnard and Thompsons algorithm (including the Moravec operator) as described above. The disparity limits set are +/- 12 pixels vertical disparity, and +/- 20 pixels horizontal disparity. The local neighborhood size used during update is of radius 12 pixels. The number of iterations during phase 3 is fixed at 10.

The first dataset consists of four synthetic image pairs used during functional testing. Each image consists of black and white squares superimposed on a grey background. In the right image the position of each square is shifted by a fixed horizontal and vertical disparity. These timings represent performance approaching the optimum that can be acheived with the current implementation, most Transputers in the chain process exactly the same number of reference features, and the number of local match sub-networks to be examined for each candidate match during the update phase is the same. As a result, the load on the processor network is balanced.

The second dataset is a simulated SPOT stereo pair. Clearly, the results obtained can only be taken as indicators of the performance of a modified algorithm processing real SPOT data. Final timings will not be available until the inadequacy of the Moravec interest operator, and the other algorithmic deficiencies indicated have been resolved and typical 'real world' datasets are available. However, selected results for this dataset should be representative of this implementation strategy applied to 'real' data i.e. the number and relative proportion of different operations performed is consistent with the expected instruction mix.

The following abbreviations have been used in the tabulation of results -

obs u	=	observed average phase 3 utilisation
anti u	=	anticipated average phase 3 utilisation = f' / max f
f'	=	observed average number of reference features per processor
max f	=	observed maximum number of features per processor
t	=	threshold applied during phase 1
w	=	suppression window size applied during phase 1
I	=	observed maximum input time (ms)
p1	=	observed maximum phase 1 time (ms)
p2	=	observed maximum phase 2 time (ms)
p3	=	observed maximum phase 3 time (ms)
p4	=	observed maximum phase 4 time (ms)
ptp	=	average processing time per reference feature point (ms)

As expected, the anticipated and observed utilisation metrics are better for the synthetic dataset than those observed for the "real" data. In most cases the anticipated utilisation provides a reasonable upper bound prediction of observed processor utilisation except when high thresholds and large window sizes are used which tend to produce extremely sparse, but clustered feature points (see Collins et al 1987). In many cases, the differences between anticipated and observed utilisation values can be attributed to the neglection of communications overhead in the predicted values.

It is clear that as the average number of features per processor increases, the utilisation of the processor increases with a corresponding decrease in the processing time per point. If the "critical" density required for convergence is higher than the densities of simulated SPOT feature points shown here (which it would appear to be based upon comparison of the average processing time per reference feature for the two datasets), it is likely that the observed processor utilisations during phase 3, and the average processing time per point, will increase towards those observed for the synthetic datasets. There are two reasons for this. Firstly, the clustering observed at high thresholds (i.e. low feature point densities) is less pronounced at lower thresholds, so the workload will be more evenly distributed across the processor network. Secondly, an increase in the density of feature points, for a fixed size neighborhood will imply that more candidate matches must be scanned for each update, implying a stretched computational cycle relative to the communications/synchronisation cycle.

A Results: Dataset 1

obs u	anti u	f'	max f	t	w	I	p1	p2	p3	p4	ptp
0.730	0.9	3.6	4	100	5	4391	647	64	604	1	18.28
0.723	0.9	14.4	16	100	5	4391	666	199	2159	3	10.51
0.714	0.9	7.2	8	100	5	4390	656	105	1035	1	12.48
0.709	0.9	10.8	12	100	5	4391	663	155	1555	2	11.00

B Results: Dataset 2

obs u	anti u	f'	max f	t	w	I	p1	p2	p3	p4	ptp
0.633	0.668	16.7	25	75	9	4391	726	382	2468	4	10.72
0.621	0.704	16.2	23	60	11	4391	752	300	2038	3	9.55
0.549	0.611	14.1	23	90	7	4391	695	220	1386	2	8.20
0.523	0.605	18.2	30	90	5	4391	791	317	3106	4	11.62
0.515	0.585	13.5	23	75	11	4391	729	253	1267	2	8.37
0.450	0.567	11.9	21	105	5	4393	676	177	1116	2	8.28
0.442	0.636	11.5	18	90	9	4391	695	175	996	2	8.16
0.385	0.609	9.8	16	90	11	4391	699	151	656	1	7.73
0.327	0.606	9.7	16	105	7	4391	676	126	692	1	7.71
0.319	0.635	8.25	13	105	9	4391	681	111	823	1	9.79
0.216	0.546	7.1	13	105	11	4394	684	107	483	1	8.98

X CONCLUSIONS

The application of parallel processing techniques based on the Transputer to stereo image matching have been described. Preliminary speed results have been reported and discussed in terms of both the algorithm and implementation used, and deficiencies identified and alternative solutions proposed. These results are encouraging, despite the fact that no assumption of epipolar geometry has been made. The utilisations and processing times presented here indicate that this algorithm can be implemented using the current strategy to satisfy the real-time requirements of processing SPOT data providing that a relatively uniform distribution of features is obtained. A flexible framework has been developed to permit alternative implementation strategies to be explored if they appear more attractive than the current technique.

ACKNOWLEDGEMENTS

The authors would also like to thank Greg Robertson, Andy Deacon and Dr. David Watson at THORN EMI Central Research Laboratories, Hugh Webber and Sally Baker at The Royal Signals and Radar Establishment, and Dr. Paul Otto and Arwyn Anthony at University College London for their help and support. during the preparation of this paper.

REFERENCES

1 Moravec, H.P., "Towards automatic visual obstacle avoidance" **Proc. 5th Int. Joint Conf. Artificial Intell.**, Cambridge, Ma, 1977, p584.

2 Barnard, S.T., Thompson, W.B., "Disparity Analysis of Images" **IEEE Transactions on Pattern Analysis and Machine Intelligence**, Vol.PAMI-2, No.4, 1980, pp 333-340.

3 Harp, J.G., Roberts, J.B.G., Ward, J.S., "Signal Processing with Transputer Arrays (TRAPs)" **Computer Physics Communications 37**, 1985, pp 77-86.

4 Chevrel, M., Curtois, M., Wiell, G., "The SPOT Satellite Remote Sensing Mission", **Photographic Engineering and Remote Sensing**, Vol.47, No.8,1981, pp 1162-1171.

5 Dowman, I.J., "An Evaluation of Simulated SPOT data", **Contract Report** to the N.E.R.C., 1982.

6 Collins K.A., Roberts J.B.G., Anthony A., Muller J-P. "Transputer Arrays for Real Time Feature Point Stereo Matching", **ISPRS Intercommision Conference on Fast Processing of Photogrammetric Data**, Interlaken Switzerland, June 1987.

7 Forstner W., "A Feature Based Correspondence Algorithm for Image Matching", **Int. Arch. of Photogrammetry**, Vol 26-III, Rovaniemi, 1986.

8 Pollard, S.B., Mayhew, J.E.W., Frisby, J.P., "PMF: A Stereo Correspondence Algorithm Using a Disparity Gradient Limit", **Perception 1985**, Vol 14, pp 449-470.