Journal of Zhejiang University SCIENCE
ISSN 1009-3095
http://www.zju.edu.cn/jzus
E-mail: jzus@zju.edu.cn

Constrained branch-and-bound algorithm for image registration*

JIN Jian-qiu (金剑秋)^{1,2,3}, WANG Zhang-ye (王章野)^{‡1}, PENG Qun-sheng (彭群生)^{1}

(*[1]State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou 310027, China*)

(*[2]College of Computer & Information Engineering, Zhejiang Gongshang University, Hangzhou 310035, China*)

(*[3]Department of Mathematics, Zhejiang University, Hangzhou 310027, China*)

E-mail: jqjin@cad.zju.edu.cn; zywang@cad.zju.edu.cn; peng@cad.zju.edu.cn

Received Feb. 2, 2005; revision accepted June 12, 2005

Abstract: In this paper, the authors propose a refined Branch-and-Bound algorithm for affine-transformation based image registration. Given two feature point-sets in two images respectively, the authors first extract a sequence of high-probability matched point-pairs by considering well-defined features. Each resultant point-pair can be regarded as a constraint in the search space of Branch-and-Bound algorithm guiding the search process. The authors carry out Branch-and-Bound search with the constraint of a pair-point selected by using Monte Carlo sampling according to the match measures of point-pairs. If such one cannot lead to correct result, additional candidate is chosen to start another search. High-probability matched point-pairs usually results in fewer loops and the search process is accelerated greatly. Experimental results verify the high efficiency and robustness of the author's approach.

Key words: Image registration, Branch-and-Bound, Constrained refinement
doi:10.1631/jzus.2005.AS0094 **Document code:** A **CLC number:** TP391.41

INTRODUCTION

Image registration is aimed at establishing the correspondence between (or among) two (or more) images of the same scene taken at different times, from different viewpoints, and/or by different sensors. Image registration is a classical problem that has wide applications in a variety of fields (Brown, 1992), such as computer vision, pattern recognition and medical image analysis, etc.

Image registration has been studied for three decades. Most registration methods contain four major components (Brown, 1992; Zitova and Flusser, 2003):

1. Feature space. Determining which feature to match is the first step in image registration. Some features such as image pixels, edges, points and regions are in common use. How to detect the feature other than image pixels is a subsequent problem. Furthermore, besides the position of the feature, can any additional attributes of the feature be provided?

2. Similarity measure. Image registration aims at establishing the best correspondence among images. It is an optimization process so the target function must be determined. The function, called similarity measure which measures the similarity between features of two images, depends on feature space. For the feature being image pixels correlation coefficient, mutual information is in regular use. For points as the feature Euclidean distance, Hausdorff distance and partial Hausdorff distance can be used.

3. Search space. It is important to determine which kind of transform would make the images matched. Search space, in which we will search the transform aligning the images, depends on application. In image based modelling, global transform does not exist that match the images taken from different viewpoints (Zhang, 1998). But if the camera field of

‡Corresponding author

*Project supported by the National Basic Research Program (973) of China (No. 2002CB312101), the National Natural Science Foundation of China (Nos. 60475013 and 60273053) and Defense Science and Technology Key Lab. Foundation of China (No. 51476070101JW0409)

view is small and the object size is small enough with respect to the distance from the camera to the object that exists in the remote sensor images, the images can be registered by an affine transformation (Zhang, 1998). In this case, the search space is an affine transformation group.

4. Search strategy. Search strategy deals with how to find the optimum transformation in the search space. The usual search strategies are Relaxation Labelling (Price, 1985), Branch-and-Bound (Mount *et al*., 1998) and RANSAC [Random Sample Consensus (Fischler and Bolles, 1981)], etc.

In this paper, we are not going to discuss feature detection in detail because it is beyond the scope of this study. Our focus is on the search strategy, and it is assumed that the feature points have been extracted. The search space is set to affine transformation group because the images to be registered are taken by remote sensor. Our problem can also be regarded as point-pattern matching problem which has wide applications in such fields as machine vision, computational biology and computational geometry.

PROBLEM FORMULATION

A general point pattern matching problem can be defined mathematically as given below:

Problem $P(d,D,\mathcal{T})$ Given two point-sets $A,B\subset\mathbb{R}^d$ and $\varepsilon>0$, find a transform $T\in\mathcal{T}$ for which $D(T(B),A)\leq\varepsilon$. Where $\mathcal{T}$ is a transformation space, D is a distance function which can be Euclidean distance, Hausdorff distance or partial Hausdorff distance.

For our application in image registration, our problem is $P(2,D,\mathcal{T})$, where $\mathcal{T}$ is an affine transformation group, and distance function D is chosen as partial Hausdorff distance.

PREVIOUS WORK

There are numerous algorithms on the point pattern matching problem. Most algorithms arising from computational geometry only address the transformation space composed of translation, rotation and scaling or its sub-space. Basic Alignment (Garder and Lawton, 1996), Multiple Grid (Indyk *et al*., 1999) and Single Grid methods described in (Gavrilov *et al*., 1999) limited their transformation space to isometric transformation. The algorithm proposed by Chang *et al.*(1997) is good for its robustness and computational efficiency, but it only deals with affine transformation without shear. Relaxation matching algorithm (Price, 1985) and RANSAC method (Fischler and Bolles, 1981) can deal with affine transformation, but they also have some shortcomings. Relaxation can be viewed as moving around in a multidimensional space, searching for the global maximum of match quality. It can correct some false initial matching derived from local features. But if the error rate of initial matching is high which usually happens in remote sensing image registration, it is difficult to obtain correct result. RANSAC method is famous for its robustness. But its search space increases rapidly with the growth of the size of the point sets.

BRANCH-AND-BOUND ALGORITHM

Our method is based on the Branch-and-Bound framework. Before the introduction of the framework, we present some key concepts below.

Partial Hausdorff distance

Partial Hausdorff distance is adopted as similarity measure in this paper due to its robustness. For two point sets A and B, their Hausdorff distance is defined by

$$H(A,B)=\max(h(A,B),h(B,A)) \tag{1}$$

where

$$h(A,B)=\max_{a\in A}\min_{b\in B}\|a-b\| \tag{2}$$

Hausdorff distance is perfect in mathematics. But it is sensitive to noise. Partial Hausdorff distance can overcome this drawback (Rucklidge, 1995). It is defined by

$$H^k(A,B)=\max(h^k(A,B),h^k(B,A)) \tag{3}$$

where

$$h^k(A,B)=K^{\text{th}}_{a\in A}\min_{b\in B}\|a-b\| \tag{4}$$

where K^{th} is the function which returns the kth largest element of the set. The distance mentioned in the rest of this paper is regarded as partial Hausdorff distance if it is not specified otherwise.

Affine transformation cell

We discuss affine transformation mainly, but the idea can also be used for other global transform. A 6D vector can denote an affine transformation. Let $\boldsymbol{a}=(a_1,a_2,\ldots,a_6)$. Its behavior on the point $P(x,y)$ is defined by

$$x'=a_1x+a_2y+a_3, \qquad y'=a_4x+a_5y+a_6 \tag{5}$$

Affine transformation cell $[\boldsymbol{l}, \boldsymbol{h}]$ is a set of affine transformations whose elements are between $\boldsymbol{l}$ and $\boldsymbol{h}$, where $\boldsymbol{l}\le\boldsymbol{h}$ ($l_i\le h_i$, i=1,2,…,6). A cell's (short for affine transformation cell) respondence on a point P is a rectangle point set whose left-upper vertex is $\boldsymbol{l}P$ and right-button vertex is $\boldsymbol{h}P$. This rectangle point set is called uncertainty region. In this way, each cell is associated with a collection of uncertainty regions. Let $d_{\mathcal{T}}(a,b)$ denote the distance between the uncertainty region $\mathcal{T}b$ and the point a which is the minimum distance between a and any part of $\mathcal{T}b$, where $a\in A$, $b\in B$ and $\mathcal{T}$ is a cell. So we can define partial Hausdorff distance $H^k(\mathcal{T})$ between point set A and uncertainty region set $\mathcal{T}B$:

$$H^k(\mathcal{T})=\max(K^{\text{th}}_{a\in A}\min_{b\in B}d_{\mathcal{T}}(a,b),K^{\text{th}}_{b\in B}\min_{a\in A}d_{\mathcal{T}}(a,b)) \tag{6}$$

In addition, $\boldsymbol{t}$ is any transformation in the cell $\mathcal{T}=[\boldsymbol{l}, \boldsymbol{h}]$ (in our experiment $\boldsymbol{t}=(\boldsymbol{l}+\boldsymbol{h})/2$). The distance between $\boldsymbol{t}B$ and A is denoted by $H^k_{hi}(\mathcal{T})$. It is obvious that $H^k_{hi}(\mathcal{T})$ is upper bound of $H^k(\mathcal{T})$.

Basic branch-and-bound algorithm

We have two point sets A and B to be registered. Branch-and-Bound algorithm searches a transformation $\boldsymbol{t}$ by cell decomposition so that the distance between A and $\boldsymbol{t}B$ is small enough (Rucklidge, 1995). The algorithm is described as follows:

Step 1: Initialize $\mathcal{T}_0$ and push $\mathcal{T}_0$ into priority queue Q. Initialize H_b and ε.

Step 2: If Q is not empty, pop $\mathcal{T}$ from Q, otherwise go to Step 6.

Step 3: Compute $H^k(\mathcal{T})$. If $H^k(\mathcal{T})>H_b$, go to Step 2.

Step 4: Compute $H^k_{hi}(\mathcal{T})$. If $H^k_{hi}(\mathcal{T})<H_b$, $H_b\leftarrow H^k_{hi}(\mathcal{T})$. If $H_b<\varepsilon$, go to Step 6.

Step 5: Decompose $\mathcal{T}$ into $\mathcal{T}_{\text{l}}$ and $\mathcal{T}_{\text{h}}$, and push the two cells into Q. Go to Step 2.

Step 6: If $H_b<\varepsilon$, the algorithm successfully end, otherwise the algorithm fails.

There are two points which need to be clarified. One is how to set the priority of the elements of the queue Q. As mentioned above, $\mathcal{T}B$ is a set of uncertainty regions. We define the size of an uncertainty region to be its longest side and define the size of a cell to be largest size among $\mathcal{T}B$. The size of a cell is chosen as the priority of the cell. The second problem is how to decompose a cell in Step 5. To determine which component to split, we determine which of the six components of the cell contributes most to the size of the uncertainty region, and then we split the cell through its midpoint along this dimension.

Bound alignment

Mount *et al.*(1998) proposed Bound Alignment method to accelerate the search of Branch-and-Bound algorithm. The method is that we check the number of the "alignable" uncertainty regions before computing $H^k_{hi}(\mathcal{T})$ in Step 4. An uncertainty region is said to be "alignable" if there is at most one point of A in the region. The point in A is denoted by $P_A(U)$ which is associated with the uncertainty region U and the point in B associated with U is denoted by $P_B(U)$. If the number of the "alignable" uncertainty regions is beyond a threshold set at the beginning, we say the cell $\mathcal{T}$ is "alignable" and perform alignment process for the cell, where the alignment process matches $P_A(U)$ with $P_B(U)$ for each "alignable" uncertainty region and from it an affine transformation can be solved. In the end, we verify whether the affine transformation is what we desired.

BRANCH-AND-BOUND WITH CONSTRAINT

The method introduced in Section 2 only uses the position information on feature point. The advantage of these methods is they have the most extensive applications. In practical application of image registration, more information can be utilized. For example, the variance and entropy of the neighbor of a point, the correlation coefficient and mutual information between the neighbors of a pair of points and so on are available in many applications. On some particular occasions, user interaction can be provided. We make use of the information on feature point besides its position in order to accelerate the search process. This information can be abstracted as match measure between pair-point.

According to these match measures, high match measure pair-points are selected by using Monte Carlo sampling. It is different from RANSAC in that for each sampling only one pair-point is selected. If a pair-point is really matched, there are some constraints in the cell. Suppose that $P(x,y)$ in B and $P'(x',y')$ in A are matched pair-point, we have the constraint on the transformation $\boldsymbol{a}$:

$$x'=a_1x+a_2y+a_3, \quad y'=a_4x+a_5y+a_6 \tag{7}$$

For a_1

$$a_1=(x'-a_2y-a_3)/x \tag{8}$$

For current cell $\mathcal{T}=[\boldsymbol{l}, \boldsymbol{h}]$, due to $l_2\le a_2\le h_2$, $l_3\le a_3\le h_3$ and the coordinates of image pixels being positive, we have

$$a_1\ge(x'-h_2y-h_3)/x, \quad a_1\le(x'-l_2y-l_3)/x \tag{9}$$

According to this, it is possible to reduce the interval $[l_1, h_1]$

$$\begin{aligned} &l_1\leftarrow\max(l_1,(x'-h_2y-h_3)/x),\\ &h_1\leftarrow\min(h_1,(x'-l_2y-l_3)/x) \end{aligned} \tag{10}$$

Similarly, for other components we have

$$\begin{aligned} &l_2\leftarrow\max(l_2,(x'-h_1x-h_3)/y),\\ &h_2\leftarrow\min(h_2,(x'-l_1x-l_3)/y)\\ &l_3\leftarrow\max(l_3, x'-h_1x-h_2y),\\ &h_3\leftarrow\min(h_3, x'-l_1x-l_2y)\\ &l_4\leftarrow\max(l_4,(y'-h_5y-h_6)/x),\\ &h_4\leftarrow\min(h_4,(y'-l_5y-l_6)/x)\\ &l_5\leftarrow\max(l_5,(y'-h_4x-h_6)/y),\\ &h_5\leftarrow\min(h_5,(y'-l_4x-l_6)/y)\\ &l_6\leftarrow\max(l_6, y'-h_4x-h_5y),\\ &h_6\leftarrow\min(h_6, y'-l_4x-l_5y) \end{aligned} \tag{11}$$

This process should be performed repeatedly until all components of the cell keep unchanged. Usually it is repeated one to three times. Thus we get a smaller cell $\mathcal{T}'$ than the original cell $\mathcal{T}$.

But the pair-point sampled is uncertainly matched in fact (the pair-point being matched with high probability). We cannot discard the part of $\mathcal{T}-\mathcal{T}'$. It is pushed into the queue Q. But $\mathcal{T}-\mathcal{T}'$ is usually not a cell. It can be expressed by the sum of some cells.

Suppose $\mathcal{T}=[\boldsymbol{l}, \boldsymbol{h}]$ and $\mathcal{T}'=[\boldsymbol{l}', \boldsymbol{h}']$. The $\mathcal{T}-\mathcal{T}'$ is expressed by the sum of $\mathcal{T}_1$, $\mathcal{T}_2$, …, $\mathcal{T}_{12}$, where $\mathcal{T}_i$ (i=1,2,…,12) is solved by the following procedure:

```
For $k$=1 to 6
    $\hat{\boldsymbol{l}}=\boldsymbol{l}$,  $\hat{\boldsymbol{h}}=\boldsymbol{h}$;
    $\hat{h}_k=l'_k$,  $\mathcal{T}_{2k-1}=[\hat{\boldsymbol{l}}, \hat{\boldsymbol{h}}]$;
    $\hat{l}_k=h'_k$, $h'_k=h_k$, $\mathcal{T}_{2k}=[\hat{\boldsymbol{l}}, \hat{\boldsymbol{h}}]$;
    $l_k=l'_k$, $h_k=h'_k$;
End
```

The priority of these cells is set to low value before they are pushed into Q. We call this process constrained refinement.

So far by incorporating basic Branch-and-Bound algorithm, Bound Alignment and constrained refinement, our method is described as follows:

Step 1: Initialize $\mathcal{T}_0$ and push $\mathcal{T}_0$ into priority queue Q. Initialize H_b and ε. Compute the match measure of any pair-point between point sets A and B which will be used for Monte Carlo sampling in Step 2.

Step 2: Get a pair-point (P', P) by using Monte Carlo sampling.

Step 3: If Q is empty, go to Step 9. Or else pop $\mathcal{T}$ from Q and check $\mathcal{T}$ whether it is with respect to the current pair-point. If yes, do resampling and get a new pair-point update (P', P).

Step 4: Perform the constrained refinement process due to the constraint of (P', P).

Step 5: Compute $H^k(\mathcal{T})$. If $H^k(\mathcal{T})>H_b$, go to Step

3.

Step 6: Check whether $\mathcal{T}$ is alignable. If yes, perform the Bound Alignment process and return a transform ***t***. Then compute the distance between ***tB*** and A. If the distance is smaller than ε, the algorithm successfully ends. Or else, go to Step 3.

Step 7: Compute $H_{hi}^{k}(\mathcal{T})$. If $H_{hi}^{k}(\mathcal{T}) < H_b$, $H_b \leftarrow H_{hi}^{k}(\mathcal{T})$. If $H_b<\varepsilon$, go to Step 9.

Step 8: Decompose $\mathcal{T}$ into $\mathcal{T}_l$ and $\mathcal{T}_h$, and push the two cells into Q. Go to Step 2.

Step 9: If $H_b<\varepsilon$, the algorithm successfully ends. Or else, the algorithm fails.

IMPLEMENT AND EXPERIMENT

As in (Mount *et al*., 1998) we use k-d tree (Exactly, we use the library ANN (library for Approximate Nearest Neighbor Searching), see http://www.cs.umd.edu/mount/ANN) method to compute the partial Hausdorff distance between two point sets. Aside from adding Monte Carlo sampling and constrained refinement to Branch-and-Bound with Bound Alignment, the implementation was the same for our method and that in the previous work so that the comparison between the two methods is objective and justice. In our experiments the process of Monte Carlo sampling is not included when timing our programs because it may have different implementation with respect to different application and its computational cost is usually low.

We compare CPU clock between our method and Bound Alignment method. The CPU clock cost by our programs strongly depends on the computer platform and concrete implementation involves choosing some thresholds, so the CPU clock listed in Table 1 is only for comparison.

SYNTHESIS EXPERIMENTS

In our synthetic experiments the point set B is randomly generated. Part of A is generated by transformation of part of A with the addition of small noise. The other part is randomly generated. In these experiments, the sequence of pair-point is selected manually instead of by Monte Carlo sampling. The series number of the first matched pair-point is denoted by *NFM*. To simulate reality application we time our programs in various NFMs. As shown in Table 1, our method improves the search efficiency significantly. Even if *NFM* is 4, our method has a big advantage over Bound Alignment in CPU clock and the times of decomposition (ToD). We also test the program under condition that *NFM* is bigger than 4. We find the overhead resulting from Constrained Refinement process is small compared to the whole program. Suppose the overhead is big (we cannot assure that the overhead is always small), we can save CPU time by taking the strategy that the constrained refinement is no longer performed if the optimum transformation has not be abstained after the several pair-points are sampled.

Table 1 The comparison between our method (constrained refinement) and Bound Alignment

Inlier	*NFM*	Constrained refinement		Bound alignment	
		CPU clock	ToD	CPU clock	ToD
1/3	1	125	84	1625	1075
	2	516	160		
	3	765	212		
	4	813	223		
1/2	1	188	113	3672	1793
	2	453	163		
	3	625	188		
	4	656	188		
2/3	1	203	118	5360	1075
	2	766	258		
	3	1125	324		
	4	1188	380		

Aerial photo image registration

In this experiment we had two aerial photo Visual/Infrared images registered. The corner points as feature points were extracted by SUSAN method (Smith and Brady, 1997). The images with the corner points are shown in Fig.1a and Fig.1b. The mutual information between the neighbors of the corner points is computed. And according to it, the match measures are computed that will be used in Monte Carlo sampling process. In this experiment the CPU time cost of our method is 0.12 s and that of Bound Alignment is 1.5 s (the computer platform was the same as that in Synthetic Experiments). The registered image is shown in Fig.1c, and Fig.1d is fusion result. In fusion stage, a pseudo-color image fusion method (Pohl and van Genderen, 1998) was used.

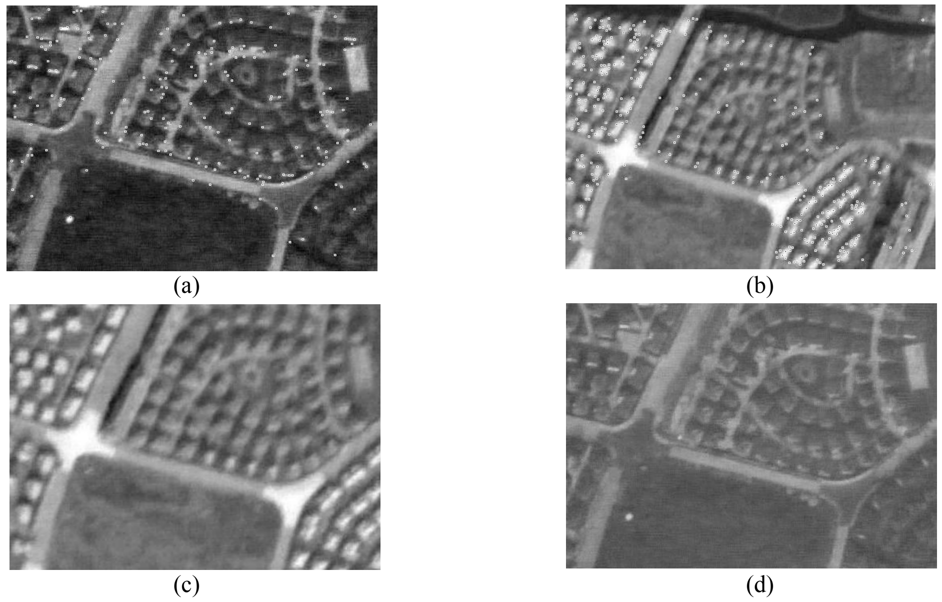

Fig.1 **Registration and fusion between visual and infrared images. (a) Visual image with corner points; (b) Infrared Image with corner points; (c) The registered image; (d) The fusion image**

CONCLUSION

Branch-and-Bound algorithm is often used in image registration and integer programming (Frederick and Gerald, 1995). But the computational complexity of the algorithm is higher than polynomial complexity. Its computational bottleneck is a great lot of cell de-compositions and the computation of partial Hausdorff distance for each cell. The constraint of pair-point can reduce the search transformation space. But the reduced space cannot be expressed in analysis, it can be discretely solved with the help of Branch-and-Bound algorithm. In this way, Constrained Refinement process is harmoniously combined with Branch-and-Bound algorithm and Bound Alignment. Our method is simple and effective.

References

Brown, L.G., 1992. A survey of image registration techniques. *ACM Computing Surveys*, **24**(4):326-376.

Chang, S.H., Cheng, F.H., Hsu, W.H., Wu, G.Z., 1997. Fast algorithm for point pattern matching: invariant to translations, rotations and scale changes. *Pattern Recognition*, **30**(2):311-320.

Fischler, M.A., Bolles, R.C., 1981. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communication of the ACM*, **24**(6):381-395.

Frederick, S.H., Gerald, J.L., 1995. Introduction to Operation Research, 6th Edition. The McGraw-Hill Press, p.675-687.

Garder, W.F., Lawton, D.T., 1996. Interactive model-based vehicle tracking. *IEEE Transaction Pattern Analysis and Machine Intelligence*, **18**(11):1115-1121.

Gavrilov, M., Indyk, P., Motwani, R., Venkatasubramanian, S., 1999. Geometric Pattern Matching: A Performance Study. Proc. 15th Annu. ACM Sympos. Comput. Geom., p.79-85.

Indyk, P., Motwani, R., Venkatasubramanian, S., 1999. Geometric Matching under Noise: Combinatorial Bounds and Algorithms. Proceedings of 10th Annual SIAM-ACM Symposium on Discrete Algorithms, p.457-465.

Price, K.E., 1985. Relaxation matching techniques–A comparison. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, **7**(5):617-623.

Mount, D.M., Nathan, S., Netanyahu, J., Moigne, L., 1998. Improved Algorithms for Robust Point Pattern Matching and Applications to Image Registration. Proceedings of the Fourteenth Annual Symposium on Computational Geometry, Minneapolis, Minnesota, United States, p.155-164.

Pohl, C., van Genderen, J.L., 1998. Multisensor image fusion in remote sensing: concepts, methods and applications. *International Journal of Remote Sensing*, **19**(5):823-854.

Rucklidge, W.J., 1995. Locating Objects Using the Hausdorff Distance. Proceeding of ICCV'95, p.457-464.

Smith, S., Brady, J., 1997. SUSAN–A new approach to low level image processing. *International Journal of Computer Vision*, **23**(1):45-78.

Zhang, Z., 1998. Determining the epipolar geometry and its uncertainty: A review. *International Journal of Computer Vision*, **27**(2):161-195.

Zitova, B., Flusser, J., 2003. Image registration methods: A survey. *Image and Vision Computing*, **21**(11):977-1000.