

Research Article

Automatic Three-Dimensional Measurement of Large-Scale Structure Based on Vision Metrology

Zhaokun Zhu,^{1,2} Banglei Guan,^{1,2} Xiaohu Zhang,^{1,2} Daokui Li,¹ and Qifeng Yu^{1,2}

¹ College of Aerospace Science and Engineering, National University of Defense Technology, Changsha 410073, China

² Hunan Provincial Key Laboratory of Image Measurement and Vision Navigation, National University of Defense Technology, Changsha 410073, China

Correspondence should be addressed to Zhaokun Zhu; warrior.zzk@gmail.com

Received 8 November 2013; Accepted 31 December 2013; Published 17 February 2014

Academic Editors: Y. Fu and G. Pedrini

Copyright © 2014 Zhaokun Zhu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

All relevant key techniques involved in photogrammetric vision metrology for fully automatic 3D measurement of large-scale structure are studied. A new kind of coded target consisting of circular retroreflective discs is designed, and corresponding detection and recognition algorithms based on blob detection and clustering are presented. Then a three-stage strategy starting with view clustering is proposed to achieve automatic network orientation. As for matching of noncoded targets, the concept of matching path is proposed, and matches for each noncoded target are found by determination of the optimal matching path, based on a novel voting strategy, among all possible ones. Experiments on a fixed keel of airship have been conducted to verify the effectiveness and measuring accuracy of the proposed methods.

1. Introduction

As science and technology develop, more and more large-scale structures come into our lives, such as bridges, tunnels, airplanes, airships, wind turbine blades, and antennas. To ensure safety and prevent potential accidents and disasters, it is necessary to conduct mechanical testing and dimensional quality monitoring for these large-scale structures, which require automatic three-dimensional measurement with high metric precision.

Deformation measurement of large-scale structures constitutes a 3D coordinate measuring problem involving large measurement ranges and high measuring accuracy. Compared to other measuring means such as strain gauge method [1, 2], surveying method [3, 4], and GPS method [5–7], photogrammetry displays a number of advantages [8]. For example, this image-based technology facilitates high-precision noncontact measurement, and it can handle very large arrays of 3D feature points simultaneously, thus making it suitable for time-constrained deformation measurement of large-scale structures.

Nonexpert user oriented vision metrology systems are well established in industrial metrology [9–13], though

application of photogrammetric measuring systems warrants further research and development to make the technology optimally qualified for deformation monitoring of large-scale structures.

Usually, in order to facilitate image measurement, retroreflective targets and highly controlled illumination conditions are employed in vision metrology, and together with the usage of coded target system [14–17], full automation of 3D measurement can be achieved. Recent developments in computer vision put emphasis upon 3D reconstruction of scene structure without the artificial targeting and controlled lighting conditions, which involves extraction of natural feature, feature description, dense matching [18–20], and so forth. These are the developing trends of photogrammetric vision metrology as well, but compared to photogrammetric vision metrology, without targeting, controlled lighting, and coded targets, so far these developments are still far from being fully automated for accurate 3D measurement of large-scale structure.

In this paper, we focus on photogrammetric vision metrology for fully automatic 3D measurement of large-scale structure; all relevant key techniques involved are studied, such as coded target, network orientation, and matching

of noncoded target. All will be described in the following sections detailedly.

2. Preliminaries

2.1. Pinhole Imaging Model. The image point is denoted by a homogeneous 3-vector \mathbf{x} , and the world point is denoted by a homogeneous 4-vector \mathbf{X} ; then the Pinhole Imaging Model can be simply expressed by

$$a\mathbf{x} = K [R | \mathbf{t}] \mathbf{X}, \quad (1)$$

where the scalar a is an arbitrary scale factor. K is a 3×3 upper triangular calibration matrix, holding the intrinsic parameters. R is a 3×3 rotation matrix and \mathbf{t} is a 3-vector representing the translation vector; R and \mathbf{t} together denote the rigid body transformation between the view and the world coordinate system. The 3×4 matrix $P = K[R | \mathbf{t}]$ is the projection matrix.

2.2. Epipolar Geometry. As shown in Figure 1, a spatial point P is projected to \mathbf{x} and \mathbf{x}' , respectively, on view S and S' , obviously, P , \mathbf{x} , \mathbf{x}' , and two optical centers C and C' are in the same spatial plane π , this is the well-known coplanarity or epipolar constraint, which is encoded in the 3×3 fundamental matrix F [21]:

$$\mathbf{x}'^T F \mathbf{x} = 0. \quad (2)$$

It can be deduced more intuitively in this way that \mathbf{x} and \mathbf{x}' are located on the epipolar lines $\mathbf{l} = F^T \mathbf{x}'$ and $\mathbf{l}' = F \mathbf{x}$, respectively, which means $\mathbf{l}^T \mathbf{x} = 0$ and $\mathbf{l}'^T \mathbf{x}' = 0$, respectively; both can be rewritten as (2).

The relative pose between view S and S' is $[R | \mathbf{t}]$, slightly different from what is stated in 2.1; the coordinate system of view S , which is C -XYZ, is now the world coordinate system. Let $[\mathbf{t}]_{\times}$ denote the skew symmetric matrix:

$$[\mathbf{t}]_{\times} = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}, \quad (3)$$

then the fundamental matrix F is

$$F = K'^{-T} [\mathbf{t}]_{\times} R K^{-1}, \quad (4)$$

where K and K' are the calibration matrices of view S and S' , respectively.

The fundamental matrix F can be considered without knowledge of the calibration matrices [21], once K and K' are known, which means view S and S' are calibrated; image points can be normalized

$$\hat{\mathbf{x}} = K^{-1} \mathbf{x}, \quad \hat{\mathbf{x}}' = K'^{-1} \mathbf{x}', \quad (5)$$

where $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}'$ are the normalized image points of \mathbf{x} and \mathbf{x}' , respectively; then the epipolar constraint (2) can be simplified to

$$\hat{\mathbf{x}}'^T E \hat{\mathbf{x}} = 0, \quad (6)$$

where E is the 3×3 essential matrix:

$$E = [\mathbf{t}]_{\times} R. \quad (7)$$

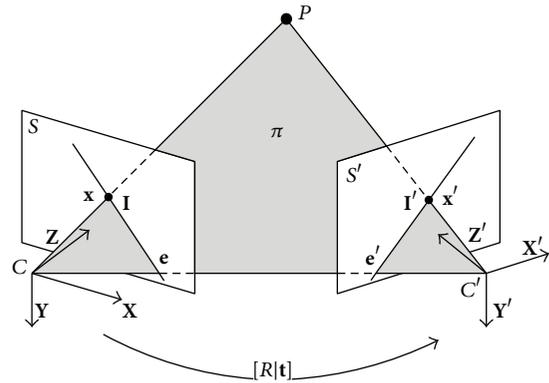


FIGURE 1: Epipolar geometry.

2.3. Relative Orientation. The mission of relative orientation is to determine the relative pose $[R | \mathbf{t}]$ between two views, as shown in (7); essential matrix E contains information about R and \mathbf{t} , so almost every existing relative orientation method, such as the 5-point method [21] and 7- [22] and 8-point methods [22], is based on recovering essential matrix first and then recovering R and \mathbf{t} from it. For simplicity and robustness, the 8-point relative orientation method is adopted in this paper [22].

The projection matrices of S and S' are $K[I | \mathbf{0}]$ and $K'[R | \mathbf{t}]$, respectively; once R and \mathbf{t} are recovered, the scene structure can be reconstructed in the coordinate system of S by spatial intersection. Preferably, bundle adjustment [23, 24] follows on when relative pose and scene structure are recovered linearly, by which optimal results can be obtained in the least squares sense. Actually, bundle adjustment serves as the last step in almost all motion and structure estimation problems [23].

3. Technical Route

Automatic deformation measurement of large-scale structure in this paper is carried out mainly in the following three steps:

- (i) automatic detection of targets, including coded and noncoded ones;
- (ii) automatic network orientation using coded targets;
- (iii) automatic matching of noncoded targets based on epipolar constraint.

Each step can be further broken down into several detailed substeps; the complete measuring scheme is shown in Figure 2. Key techniques involved are presented in detail in the following sections.

4. Design of Coded Target

Coded targets are essential to automation of the whole measuring procedure; they provide point correspondences automatically [25] and ensure reliable results in succeeding network orientation once there is a sufficient number of pairs of homologous points between two images to facilitate relative orientation [25].

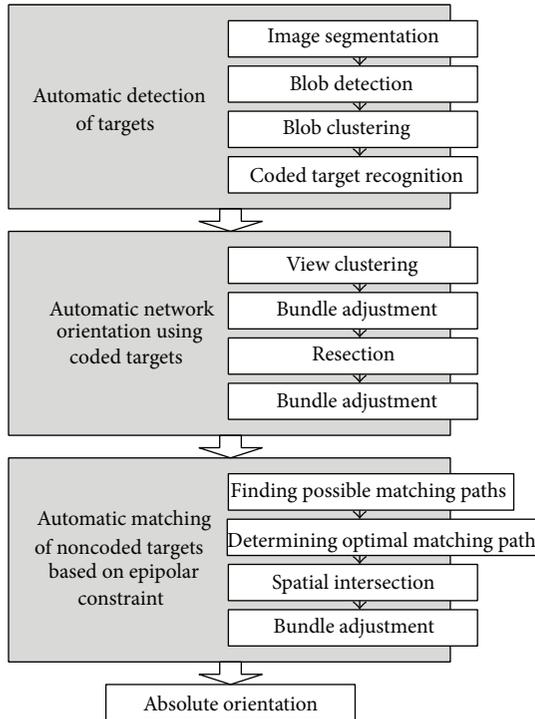


FIGURE 2: Measuring scheme.

Coded target is an automatically detected and recognized pattern of geometric features within the image [14–17]; a qualified coded target is supposed to have sufficient coding capacity and be projective invariant to a certain degree, its corresponding detection algorithm is supposed to be simple, efficient, and robust; moreover, due to measuring accuracy concern, its image points should be able to be measured with high precision.

The coded target designed in this paper is shown in Figure 3; it consists of several circular retroreflective points, which are arranged according to a certain coding pattern. One reason why circular retroreflective points are adopted is that compared with other geometric features a circular retroreflective point is always imaged as a bright blob under illumination; it is more robust against image degrading processes (e.g., defocusing or changing of the imaging distance and angle) [14]. The other reason is that noncoded target adopted in this paper consists of single circular retroreflective point, like points in coded target; it is also imaged as a bright blob under illumination; thus, identical blob detection algorithm can be applied to coded and noncoded targets at the same time in preliminary stage of detection, avoiding developing different algorithms, respectively.

Out of consideration for detection efficiency, if color images are applied, preferably, the circular retroreflective points in color can be used, just as shown in Figure 3; color information will greatly facilitate the image segmentation, which is essential to detection efficiency.

4.1. Coding Principle. There are two kinds of circular points in the designed coded target; one is reference points; there

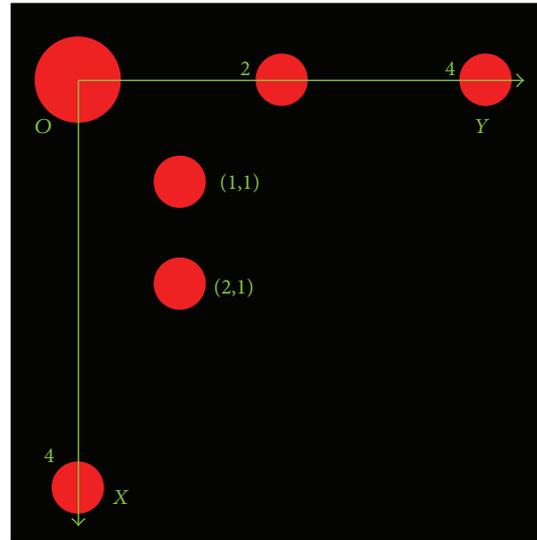


FIGURE 3: Design of coded target.

are 4 of them in total, and they together establish the two-dimensional target coordinate system $O-XY$, as shown in Figure 3; the biggest among them represents the origin O two of the rest represent two different points on Y -axis, whose coordinates are $(0, 2)$ and $(0, 4)$, respectively, and the last one represents the point on X -axis with coordinates $(4, 0)$.

The other kind is coding points; they are located in positions with positive integer coordinates (i, j) , and each one of them represents one binary bit k ; the presence and absence of it represent the status 1 and 0 of this binary bit, respectively. If there are m coding points at most per column, in other words, the X coordinate i is not more than m , then the relationship between the binary bit k and the coordinates (i, j) is given by

$$k = (j - 1) \times m + i. \tag{8}$$

Given that the capacity of coding points per row is n , the total amount of binary bits encoded by the designed coded target is $m \times n$; then the overall coding capacity is $2^{m \times n}$, which means $2^{m \times n}$ instances of coded target with different coding information can be derived, for m and n are chosen arbitrarily; thus, theoretically, there is no upper limit to the coding capacity of the designed coded target.

Take the coded target shown in Figure 3, for example; given that m and n are both 3, then it represents a binary code with $9 = 3 \times 3$ binary bits; thus, its coding capacity is 512. There are two coding points in it, with coordinates $(1, 1)$ and $(2, 1)$, respectively; according to (8), they represent the 1st and 2nd binary bits, respectively; the absence of coding points in other 7 positions with positive integer coordinates means the statuses of other 7 binary bits are 0; thus, the encoded binary code is 00000011, and the corresponding decimal code is $2^0 + 2^1 = 3$.

5. Detection of Coded and Noncoded Target

Detection of coded and noncoded targets in this paper is carried out mainly in the following four steps:

- (i) image segmentation simply based on gray information and, preferably, color information, if applicable;
- (ii) blob detection based on (Normalized Laplacian of Gaussian) NLoG;
- (iii) blob clustering based on image distance between two points; the clusters containing only single blob are determined to be noncoded targets, whereas the clusters containing the required amount of blobs are treated as potential coded targets in succeeding recognition process;
- (iv) coded target recognition based on geometric characteristics and coding principle of the designed coded target; potential coded targets cannot be diagnosed until all constraints are met in this recognition process.

5.1. *Blob Detection.* The blob detector adopted in this paper is the well-known scale invariant (Normalized Laplacian of Gaussian) NLoG, which is given as follows:

$$NLoG(x, y; \sigma) = \frac{x^2 + y^2 - 2\sigma^2}{2\pi\sigma^4} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right), \quad (9)$$

where σ is the scale parameter; and the detection of blobs with multiple scales is to detect scale-space maxima/minima, which are points that are simultaneously local maxima/minima with respect to both space and scale [26, 27].

5.2. *Blob Clustering.* As stated before, a coded target is imaged as a group of blobs, whereas a noncoded one is imaged as one single blob, which means the blobs within a coded target distributes much denser than blobs of multiple noncoded targets do; this will not change as long as the distribution density of multiple noncoded targets in space does not exceed the one of points within a coded target, and the depths of noncoded targets, relative to camera, do not vary too much from the one of a coded target. Thus, simple blob clustering based on image distance between two points can be conducted to distinguish between coded and noncoded targets.

The blob clustering process is demonstrated in Figure 4. The group of blobs surrounded by the big green circle is a coded target, and the rest blobs are all noncoded ones. The white circle surrounding each blob with radius r denotes the adjacent area of each blob; if a blob is within this adjacent area of another blob, then these two blobs are adjacent to each other and can be clustered into the same cluster. As you can see, there are no other blobs within the adjacent area of each noncoded target, so the cluster of each noncoded target contains only its own blob.

Let us check out step by step how clustering works out for coded target; if it starts at blob a first, blob b will be added into cluster next, for it is within the adjacent area of blob a ,

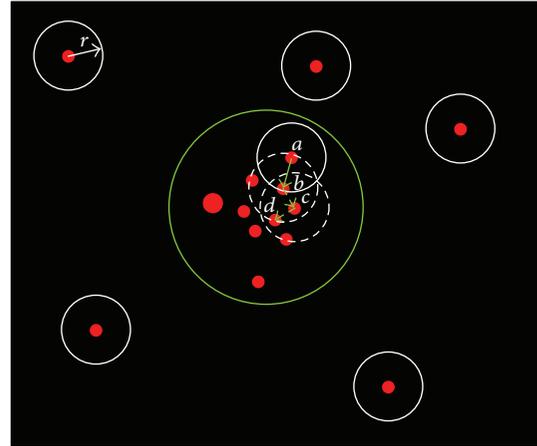


FIGURE 4: Blob clustering process.

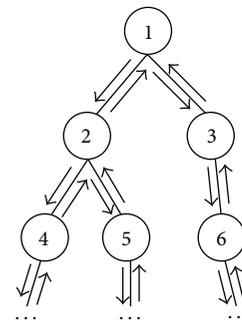


FIGURE 5: Recursive routine.

and then come blobs c and d ; clustering will keep going until no more adjacent blobs are found.

This blob clustering process, like the view clustering process which is going to be described in the succeeding section, is a standard recursive process; they both can be implemented by the recursive routine shown in Figure 5.

After clustering, clusters containing only single blob are determined to be noncoded targets directly, and clusters containing reasonable amount of blobs are determined to be potential coded targets, since a true coded target contains at least 4 reference points and limited amount of coding points. If a cluster contains unreasonable amount of blobs, blobs within it will be sentenced to be noncoded targets as well.

5.3. *Coded Target Recognition.* As stated before, coded target recognition is based on geometric characteristics and coding principle of the designed coded target; potential coded targets coming from blob clustering cannot be diagnosed until all constraints are met in this recognition process. Coded target recognition is carried out mainly in the following two steps:

- (i) recognition of reference points;
- (ii) decoding.

5.3.1. *Recognition of Reference Points.* The first thing to do is to determine which blob is the origin O of the target coordinate system. As shown in Figure 6, the origin O is much bigger than any of the other blobs, so it can be located simply

by finding the blob with the biggest size. Naturally, radius can be considered; it is a measure of blob size, and it is readily acquired in blob detection stage, but radius alone cannot ensure clear distinction between the origin O and other blobs, since the radius difference between two blobs may be smaller than the scale resolution in blob detection, which is the scale step between two NLoG templates; thus, more robust method is needed. In this paper, the grayscale sum of pixels within blob radius, in other words, the grayscale weighted area, is used to make a clear distinction of the origin O from other blobs; the blob with the biggest grayscale weighted area is determined to be the origin O .

The rest three reference points, x , y_1 , and y_2 , have their own geometric characteristics; the angle between segments Oy_1 and Ox equals the angle between Oy_2 and Ox , which is both α_0 as shown in Figure 6; besides, these two angles are bigger than angle α_i formed by any other two blobs toward origin O ; based on these, x , y_1 , and y_2 are determined in this paper by finding angle α_0 .

5.3.2. *Decoding.* Technically, the relationship between the image coordinates of a coding point, which is (x, y) , and its target coordinates (i, j) is a projective transformation; however, 4 pairs of correspondences between image and target generated by 4 reference points are unable to solve this projective transformation, since there are 3 reference points which are collinear, which are O , y_1 , and y_2 , respectively. Fortunately, when target depth, relative to camera, is far greater than target size, as is usually the case in practice, the relationship between (x, y) and (i, j) approximates to an affine transformation H_a :

$$\begin{pmatrix} j \\ i \\ 1 \end{pmatrix} = \begin{pmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}. \quad (10)$$

At least 3 pairs of correspondences, generated by 3 non-collinear points, are adequate to solve H_a ; thus, the recognized 4 reference points in Section 5.3.1 can be used to solve H_a .

Once H_a is recovered, the target coordinates (i, j) of each coding point can be computed according to (9), and corresponding binary bit k of each coding point can be further acquired according to (8); then the binary and decimal code is finally decoded.

6. Automatic Network Orientation

The mission of automatic network orientation is to automatically determine the relative pose between each view and certain reference frame, which is usually the coordinate system of the first view, and reconstruct all coded targets at the same time. A three-stage automatic network orientation strategy is proposed in this paper:

- (i) view clustering based on relative orientation with the help of coded targets and reconstruction of coded targets at the same time;
- (ii) connecting those view clusters which contain multiple views using absolute orientation;

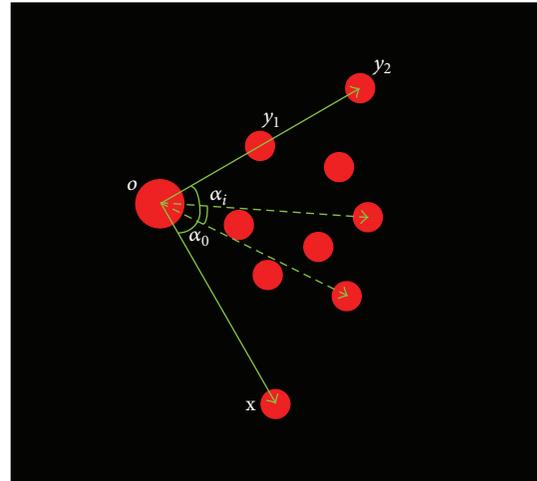


FIGURE 6: Recognition of reference points.

- (iii) conducting resections for isolate views, which cannot be clustered with other views, using coded targets reconstructed in view clustering.

6.1. *View Clustering.* As stated in [28], many researches on structure and motion recovery have been based on some certain image ordering, usually in chronological order if image set is sequential; this image ordering allows small baseline matching algorithms to be used between consecutive frames of the sequence and avoids wide baseline situations, in which matching is difficult; thus, all images can be successfully sewed together into tracks image by image.

In our case, the usage of coded targets greatly alleviates our concerns with matching; it can provide reliable point correspondences between views, even in wide baseline situations for coded target itself is robust to large perspective distortions, as stated in Sections 4 and 5. Yet this ordering remains advantageous to our case, since the small baseline conditions it brings mean bigger overlapping area between footprints of consecutive views, in other words, more point correspondences available for relative orientation. But sometimes this ordering is not guaranteed in our case, in which way relative orientation between views may fail for overlapping area between footprints of these views is not big enough to provide adequate correspondences; in order to cope with this situation and automate the network orientation process, view clustering is carried out.

Unlike view clustering in [28], in which feature matching is involved, view clustering in our case is much simpler due to the usage of coded targets; whether two views can be clustered into one group or not simply depends on whether the relative orientation between these two views succeeds or fails.

As stated in Section 5.2, like the blob clustering, view clustering process is also implemented by the recursive routine shown in Figure 5.

The whole network orientation process is demonstrated in Figure 7 by images of a steel structure, as shown; there are 9

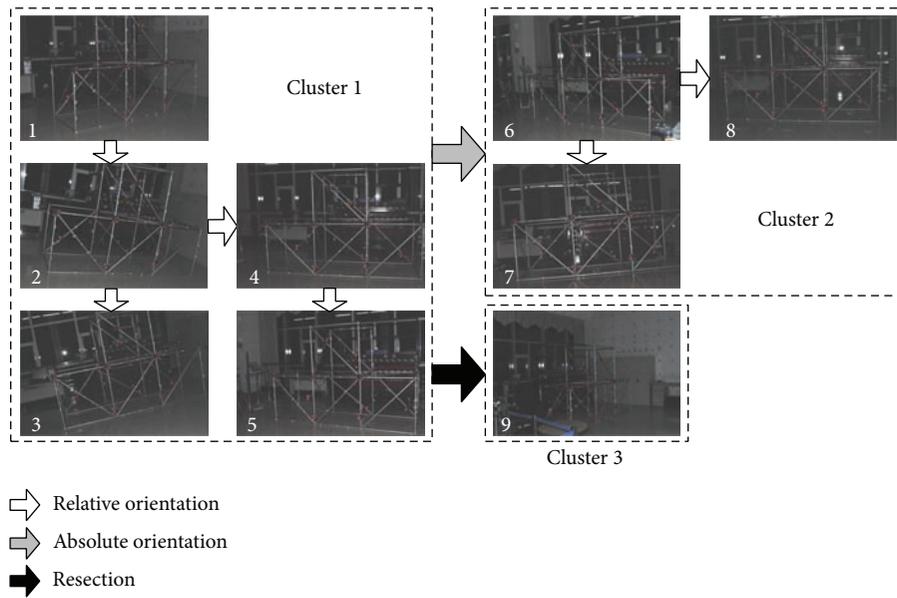


FIGURE 7: Network orientation process.

views in total, and three clusters emerge after view clustering. The first five are clustered together to be cluster 1, view 6 to view 8 form the cluster 2, and view 9 is left alone to be cluster 3.

If at least three homologous coded targets are reconstructed both in cluster 1 and cluster 2, then cluster 2 can be connected to cluster 1 using absolute orientation.

As for view 9, it is an isolate view for it observes only five coded targets, which is insufficient to carry out relative orientation between it and any other views, but connection can still be built between it and cluster 1 by resection, as long as there are adequate observed coded targets, which have already been reconstructed in cluster 1, the minimum amount required for resection can be at least 4 under certain circumstances.

Be aware that bundle adjustment is recommended as the last step in all three stages of network orientation.

7. Matching of Noncoded Target

Unlike coded target, noncoded target cannot provide its matches over multiple views by itself; besides, feature descriptor based methods [18–20] are not suitable for matching of noncoded targets in our case, where the illumination is highly controlled to ensure that all targets can be readily detected; for the same reason, images in our case are usually textureless, which makes feature descriptor not applicable.

Given that all views that can be oriented have already been automatically oriented in network orientation, noncoded targets can be matched over views based on epipolar constraints, introduced in Section 2.2, and this is also how it is done in this paper but in a subtler way. The concept of matching paths is proposed, and matching of noncoded targets in this paper is carried out mainly in two steps:

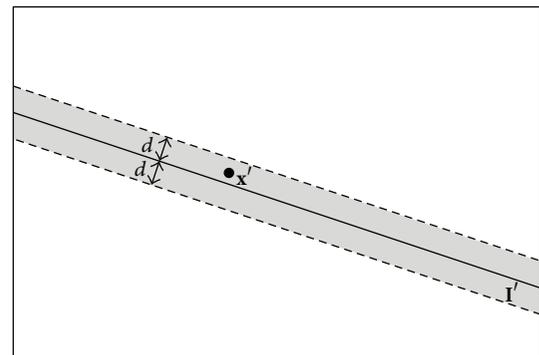


FIGURE 8: Determining potential match based on epipolar constraint.

- (i) finding all possible matching paths through multiple views for each noncoded target;
- (ii) determining the optimal matching path for each noncoded target by a novel voting strategy.

7.1. Finding Possible Matching Paths. As stated in Section 2.2 and shown in Figure 1, given the fundamental matrix F , the match in view S' , which is x' , for a given x in view S , is theoretically located on the epipolar line $l' = Fx$, which means $l'^T x' = 0$, and this is what our matching strategy is based on. But when the estimate for F and the measurements of x and x' are not error free, as is the case in practice, x' will not be exactly on l' anymore but close to it; in other words, $l'^T x' \neq 0$; thus, in practice once the image distance between x' and l' is smaller than certain threshold d , as shown in Figure 8, x' will be determined to be a potential match for x .

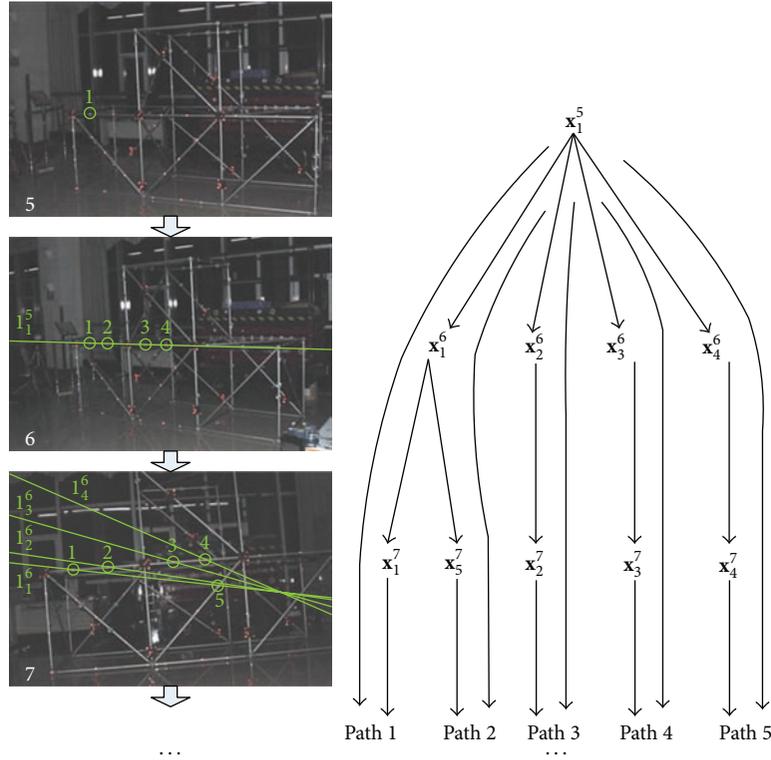


FIGURE 9: Finding possible matching paths.

Sometimes, besides P there will be other spatial points in the plane π as well, as shown in Figure 1; this will bring ambiguity to finding match for \mathbf{x} , since projections of these points on view S' are also on I' . In such cases, when we try to determine which image point on S' is the match for \mathbf{x} based on criterion shown in Figure 8, we will be expected to find multiple candidates rather than one, just as shown in Figure 9; four potential matches in view 6 for point 1 in view 5 are found.

Obviously, if there is a third view with optical center out of the plane π , this ambiguity will be eliminated. But in order to reduce complexity and increase the automaticity and robustness, we do not explicitly attempt to find this third view during matching in this paper; instead, a simpler and subtler way is adopted. We first maintain the ambiguity and find all possible matching paths through multiple views for a given point, then determine the optimal path based on a novel voting strategy, eliminating the ambiguity, and finally find the real match.

A matching path is defined as follows:

$$\text{path}(\mathbf{x}_{j_1}^{i_1}, \mathbf{x}_{j_2}^{i_2}, \dots, \mathbf{x}_{j_k}^{i_k}, \dots, \mathbf{x}_{j_w}^{i_w}), \quad (11)$$

$$1 \leq k \leq w, \quad 1 \leq i_k \leq m, \quad 1 \leq j_k \leq n(i_k),$$

where $\mathbf{x}_{j_k}^{i_k}$ denotes the k th matching image point in the matching path, i_k is the view number of this point, or in which view it is measured, j_k is its point number among all observations in view i_k , w is the total amount of points contained in the path, m denotes the total amount of views,

and $n(i_k)$ represents the total amount of observations in view i_k .

The process of finding possible matching paths for a given image point is demonstrated in Figure 9 still by images of a steel structure. As shown, the point 1 in view 5, which is \mathbf{x}_1^5 , finds four matches in view 6, resulting in four separate paths; then we continue to find matches in view 7 for those four points. Since the point 1 in view 6, which is \mathbf{x}_1^6 , can find two different matches in view 7, a previous path now splits into two, resulting in five separate paths in total, which are

$$\begin{aligned} &\text{path 1}(\mathbf{x}_1^5, \mathbf{x}_1^6, \mathbf{x}_1^7) \\ &\text{path 2}(\mathbf{x}_1^5, \mathbf{x}_1^6, \mathbf{x}_5^7) \\ &\text{path 3}(\mathbf{x}_1^5, \mathbf{x}_2^6, \mathbf{x}_2^7) \\ &\text{path 4}(\mathbf{x}_1^5, \mathbf{x}_3^6, \mathbf{x}_3^7) \\ &\text{path 5}(\mathbf{x}_1^5, \mathbf{x}_4^6, \mathbf{x}_4^7). \end{aligned} \quad (12)$$

Given that F_b^a is the fundamental matrix between view a and view b , then the epipolar line I_j^a in view b , as shown in Figure 9, is

$$I_j^a = F_b^a \mathbf{x}_j^a, \quad (13)$$

where \mathbf{x}_j^a is the point j in view a . In short, I_j^a is an epipolar line induced by \mathbf{x}_j^a .

As the process described above continues, the amount of matching paths will be expected to increase, but if the amount of paths remains one eventually, image points contained in the only matching path will naturally be determined to be matched or the voting strategy, which is described in the next section, will step in.

7.2. Determining Optimal Matching Path. The novel voting strategy proposed in this paper is still based on epipolar constraint; if image points contained in a matching path are really matched, epipolar constraint should be met between all N-choose-2 points; thus, we can determine whether a path is the real one simply by checking if all the N-choose-2 cases in it meet the epipolar constraint. However, due to error, probably not all the cases are qualified, so in practice it is reasonable to choose the path with most qualified cases instead of all qualified cases as the optimal matching path, which contains points really matched.

The vote that a matching path can get is given by

$$\text{vote} = \sum_{k_1=1}^w \sum_{k_2=1}^w \text{check}(\mathbf{x}_{j_{k_1}}^{i_{k_1}}, \mathbf{x}_{j_{k_2}}^{i_{k_2}}), \quad k_2 \neq k_1, \quad (14)$$

where $\mathbf{x}_{j_{k_1}}^{i_{k_1}}$ and $\mathbf{x}_{j_{k_2}}^{i_{k_2}}$ are, respectively, the k_1 th and k_2 th points in the path, and the function check is given as follows:

$$\text{check}(\mathbf{x}_{j_{k_1}}^{i_{k_1}}, \mathbf{x}_{j_{k_2}}^{i_{k_2}}) = \begin{cases} 1 & \text{if } \text{dist}(\mathbf{x}_{j_{k_1}}^{i_{k_1}}, F_{i_{k_1}}^{i_{k_2}} \mathbf{x}_{j_{k_2}}^{i_{k_2}}) < d \\ 0 & \text{else,} \end{cases} \quad (15)$$

where $\text{dist}(\mathbf{x}_{j_{k_1}}^{i_{k_1}}, F_{i_{k_1}}^{i_{k_2}} \mathbf{x}_{j_{k_2}}^{i_{k_2}})$ is the distance between point $\mathbf{x}_{j_{k_1}}^{i_{k_1}}$ and the epipolar line $F_{i_{k_1}}^{i_{k_2}} \mathbf{x}_{j_{k_2}}^{i_{k_2}}$ and d is some certain distance threshold, as explained in Section 7.1. In short, the function check means that if the k_1 th point is close enough to the epipolar line in view i_{k_1} induced by the k_2 th point, the path gets one vote.

Apparently, each N-choose-2 case has two votes to give, which is $\text{check}(\mathbf{x}_{j_{k_1}}^{i_{k_1}}, \mathbf{x}_{j_{k_2}}^{i_{k_2}})$ and $\text{check}(\mathbf{x}_{j_{k_2}}^{i_{k_2}}, \mathbf{x}_{j_{k_1}}^{i_{k_1}})$, respectively, so if all N-choose-2 cases in a path meet the epipolar constraint, this path will get a unanimous vote, which is $w(w-1)$.

Taking the three images shown in Figure 9 as an example again and using the voting strategy described above, path 1 gets the most votes, which is 6, whereas all other paths get only 4 votes, so path 1 is chosen to be the optimal matching path up to this point, just as what it really is; thus, the points contained in path 1 are determined to be matched.

8. Experimental Results

The main types of airship are nonrigid blimps, semirigid airship, and rigid airship. Unlike the nonrigid blimps, semirigid airship usually has a fixed keel besides internal pressure, which runs the length of the ship along the bottom of the hull, as shown in Figure 10, and provides greater structural strength to maintain its shape and structural integrity.

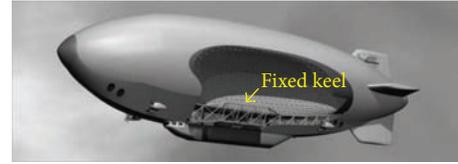


FIGURE 10: Semirigid airship.

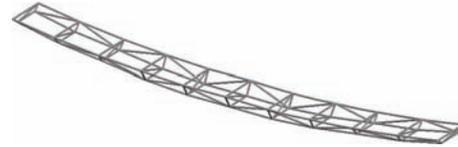


FIGURE 11: Designed keel structure.



FIGURE 12: Designed keel structure in field.

A designed keel structure in a practical semi-airship is shown in Figures 11 and 12.

In order to ensure safety and prevent potential structural failure in extreme conditions like storm, it is necessary to conduct mechanical testing and high-precision dimensional quality monitoring for this designed keel structure before the airship is put into use.

The proposed methods in this paper have been applied to this mechanical testing, and results are shown in the following figures. The reference frame is established by the three points on the chessboard shown in Figure 14, with origin located on the top left corner of the chessboard. Red noncoded targets are attached to the main axis of the keel node by node, with the positional changes of these noncoded targets reflecting the deformations of the keel under different loads and internal pressures. Red coded targets are scattered evenly in the scene to facilitate network orientation; examples of detection of coded targets are given in Figure 13, and as you can see, detection of coded targets is quite robust to large perspective distortions.

There are 18 images which are taken under one working condition; some of these images are shown in Figure 14, and the recovered camera motion and scene structure are shown in Figure 15.

In order to verify the measuring accuracy, vertical deformations of one node under different load and internal pressure conditions are measured by total station besides our automatic vision metrology as true values, and the deviations of our measurement results from the true values are given in Table 1. As you can see, measuring errors under all working conditions are within 1 mm, which is considerably good.

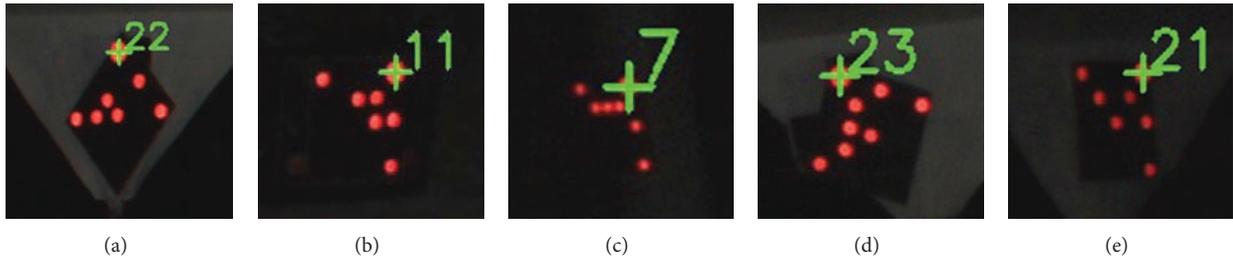


FIGURE 13: Detection of coded targets.

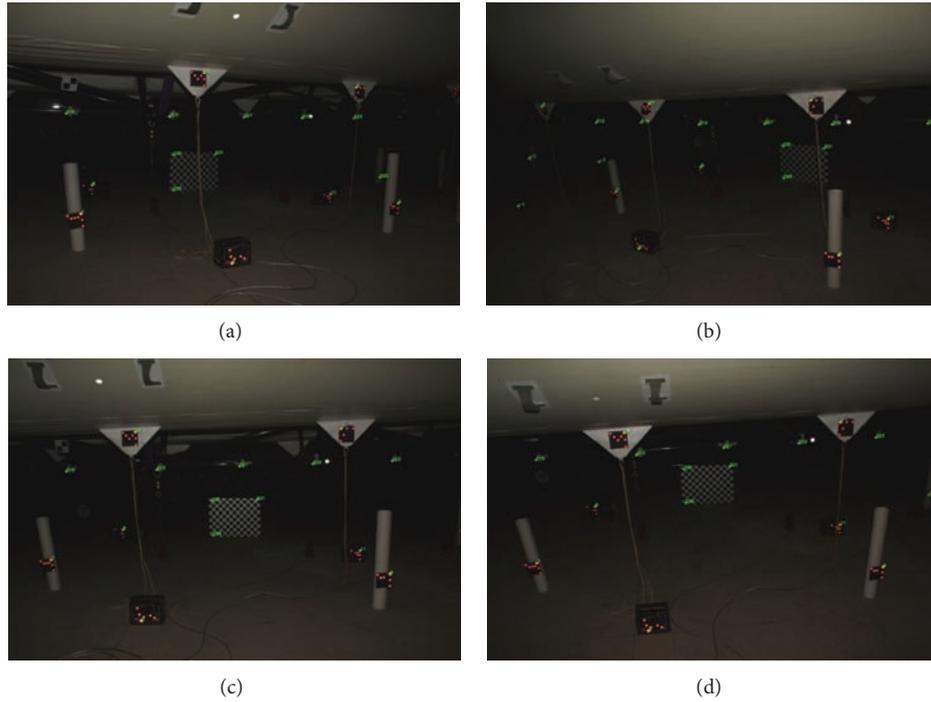


FIGURE 14: Images of scene structure.

TABLE 1: Measurement deviations.

Working conditions	1 kN/500 Pa	1 kN/700 Pa	2 kN/500 Pa	2 kN/700 Pa	3 kN/500 Pa	3 kN/700 Pa
Total station	21.78 mm	21.62 mm	46.05 mm	42.69 mm	66.42 mm	61.83 mm
Ours	22.07 mm	21.90 mm	46.61 mm	43.27 mm	67.22 mm	62.52 mm
Error	0.29 mm	0.28 mm	0.56 mm	0.58 mm	0.80 mm	0.69 mm

9. Summary and Conclusions

In this paper, all relevant key techniques involved in photogrammetric vision metrology for fully automatic 3D measurement of large-scale structure are studied. A new kind of coded target consisting of circular retroreflective discs is designed, and corresponding detection and recognition algorithms based on blob detection and clustering are presented. Then a three-stage strategy starting with view clustering is proposed to achieve automatic network orientation. As for matching of noncoded targets, the concept of matching path is proposed, and matches for each noncoded target are found

by determination of the optimal matching path, based on a novel voting strategy, among all possible ones. Experiments on a fixed keel of airship have been conducted to verify the effectiveness and measuring accuracy of the proposed methods.

Future work will mainly focus on developing new blob clustering method that is more adaptive to coded targets in different depths, and improved method that is less sensitive to starting point should be developed for finding all possible matching paths, since, in current way, different starting points may lead to different amount of points within the optimal paths with some matched points missed.

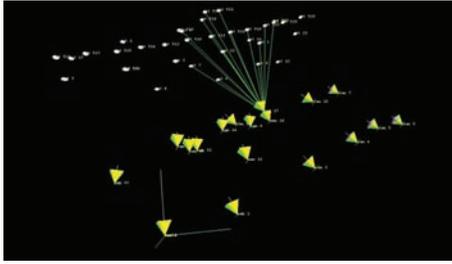


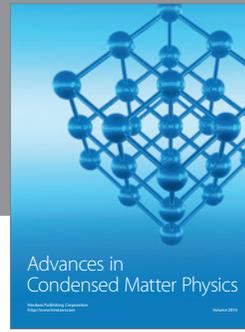
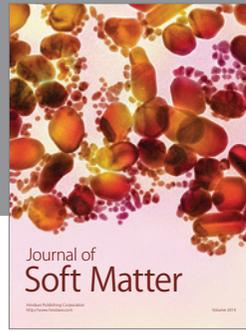
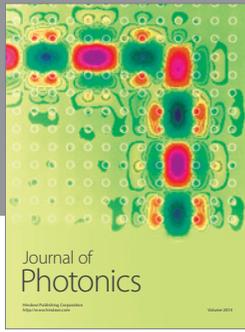
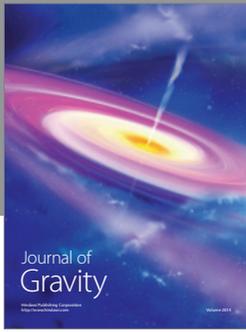
FIGURE 15: Recovery of camera motion and scene structure.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] A. L. Window and G. S. Holister, *Strain Gauge Technology*, Applied Science, 1982.
- [2] M. Bertrand, J. Meunier, M. Doucet, and G. Ferland, "Ultrasonic biochemical strain gauge based on speckle tracking," in *Proceedings of the IEEE Ultrasonics Symposium*, pp. 859–863, IEEE, October 1989.
- [3] W. F. Caspary and J. M. Rüeger, *Concepts of Network and Deformation Analysis*, University of New South Wales, 1987.
- [4] K. L. Feigl, D. C. Agnew, Y. Bock et al., "Space geodetic measurement of crustal deformation in central and southern California, 1984–1992," *Journal of Geophysical Research*, vol. 98, no. 12, pp. 21677–21712, 1993.
- [5] K. M. Larson and D. C. Agnew, "Application of the global position system to crustal deformation measurement 1. Precision and accuracy," *Journal of Geophysical Research*, vol. 96, no. 10, pp. 16547–16565, 1991.
- [6] Q. Wang, P.-Z. Zhang, J. T. Freymueller et al., "Present-day crustal deformation in China constrained by global positioning system measurements," *Science*, vol. 294, no. 5542, pp. 574–577, 2001.
- [7] J. W. Lovse, W. F. Teskey, G. Lachapelle, and M. E. Cannon, "Dynamic deformation monitoring of tall structure using GPS technology," *Journal of Surveying Engineering*, vol. 121, no. 1, pp. 35–40, 1995.
- [8] T. Luhmann, S. Robson, S. Kyle, and I. Harley, *Close Range Photogrammetry: Principles, Methods and Applications*, Whittles, 2006.
- [9] C. S. Fraser and S. Cronk, "A hybrid measurement approach for close-range photogrammetry," *Journal of Photogrammetry and Remote Sensing*, vol. 64, no. 3, pp. 328–333, 2009.
- [10] G. Ganci and H. Handley, "Automation in videogrammetry," *International Archives of Photogrammetry and Remote Sensing*, vol. 32, pp. 53–58, 1998.
- [11] C. Fraser, S. Cronk, and H. Hanley, *Close-Range Photogrammetry in Traffic Incident Management*, 2008.
- [12] C. Fraser, "Automated vision metrology: a mature technology for industrial inspection and engineering surveys," *Australian Surveyor*, vol. 46, no. 1, pp. 5–11, 2001.
- [13] C. S. Fraser, D. Brizzi, and A. Hira, *Vision-Based, Multi-Epoch Deformation Monitoring of the Atrium of Federation Square*, 2003.
- [14] S. J. Ahn, W. Rauh, and S. I. Kim, "Circular coded target for automation of optical 3D-measurement and camera calibration," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 15, no. 6, pp. 905–919, 2001.
- [15] S. J. Ahn and M. Schultes, "A new circular coded target for the automation of photogrammetric 3D-surface measurements," in *Optical 3-D Measurement Techniques IV*, A. Gruen and H. Kahmen, Eds., pp. 225–234, Herbert Wichmann, 1997.
- [16] F. Gaspard, S. Naudet, E. Noirfalise, and P. Sayd, "Coded target and photogrammetry method using such targets," Google Patents, 2008.
- [17] A. K. Vladimir and S. Alexander, *Non-Contact 3D Model Reconstruction Using Coded Target*, 1998.
- [18] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [19] H. Bay, A. Ess, T. Tuytelaars, and L. van Gool, "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.
- [20] E. Tola, V. Lepetit, and P. Fua, "DAISY: an efficient dense descriptor applied to wide-baseline stereo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 815–830, 2010.
- [21] D. Nistér, "An efficient solution to the five-point relative pose problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–770, 2004.
- [22] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, vol. 2, Cambridge University Press, 2003.
- [23] M. I. A. Lourakis and A. A. Argyros, "SBA: a software package for generic sparse bundle adjustment," *ACM Transactions on Mathematical Software*, vol. 36, no. 1, article 2, 2009.
- [24] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—a modern synthesis," in *Vision Algorithms: Theory and Practice*, B. Triggs, A. Zisserman, and R. Szeliski, Eds., vol. 1883 of *Lecture Notes in Computer Science*, pp. 298–372, Springer, 2000.
- [25] C. S. Fraser, "Network orientation models for image-based 3D measurement," *International Archives of Photogrammetry and Remote Sensing & Spatial Information Science*, vol. 36, no. 5, p. 9, 2005.
- [26] T. Lindeberg, "Scale-space theory: a basic tool for analyzing structures at different scales," *Journal of Applied Statistics*, vol. 21, 1-2, pp. 225–270, 1994.
- [27] T. Lindeberg, "Feature detection with automatic scale selection," *International Journal of Computer Vision*, vol. 30, no. 2, pp. 79–116, 1998.
- [28] F. Schaffalitzky and A. Zisserman, "Multi-view matching for unordered image sets, or 'How do I organize my holiday snaps?'," in *Computer Vision—ECCV 2002*, A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, Eds., vol. 2350 of *Lecture Notes in Computer Science*, pp. 414–431, Springer, 2002.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

