Classifying Junctions by Vector Quantization

A. Bubel, R. Bergevin
Computer Vision and Systems Laboratory, Department of Electrical and Computer Engineering
Laval University, Ste-Foy (Qc), Canada G1K7P4
[abubel, bergevin]@gel.ulaval.ca
http://www.gel.ulaval.ca/~vision
Tel: (418) 656-2131 poste 4848
Fax: (418) 656-3594

Abstract

In this paper, we present a new method for validating and classifying junctions from unlinked edge points. A scheme is proposed where junction candidates detected by an existing method indicate regions of interest.The associated branches are then detected using a binary splitting procedure on the vectors issued from edge points. Each element of these vectors is the resulting convolution of the intensity image with an oriented edge detector. Finally, the junction localization is refined and the junction type is determined. The whole process takes part in a global project which aims at developing a vision system for the recognition of significant 3D objects in a 2D image of a complex scene of man-made objects.

Index Terms

Junction classification, junction validation, binary splitting, vector quantization, edge points, object recognition, computer vision.

1. Introduction

The realization of a self-governing system such as a mobile and intelligent robot evolving in a complex environment of man-made objects is a real-world application widely studied in computer vision.

The analysis of typical real-world scenes reveals the large number as well as the diversity of involved objects and leads us to privilege a generic object recognition method. Thus, we aim at describing 3D objects according to geometric criteria which subscribe to the recognition-by-components paradigm of Biederman [2]. For instance, Bergevin & Levine [1] adopt this paradigm and describe models of man-made objects by geons. A classification of edges into straight and curved segments offers a good transition from the primary data to the description under this type of model.

A specific aspect of this global project lies in the robust extraction of edges in the intensity image. These provide the initial data for the segmentation and approximation into straight and curved segments. Our objective here is not to create a new edge detector. It is rather to develop a robust of selecting and linking edge points in order to provide a sufficiently complete description of the scene by edges.

Several authors already showed limitations of many edge detectors which produce gaps or false edges nearby junctions [4]. This often impedes the complete recovery of edges of 3D objects. It is important to notice that junctions offer a good indication of the presence and shape of a 3D object in the scene. They are especially worthwhile since they can provide in a direct way breakpoints for contour segmentation. These arguments point out the contribution of simultaneously detecting edge and junction points in order to reconstruct of curves.

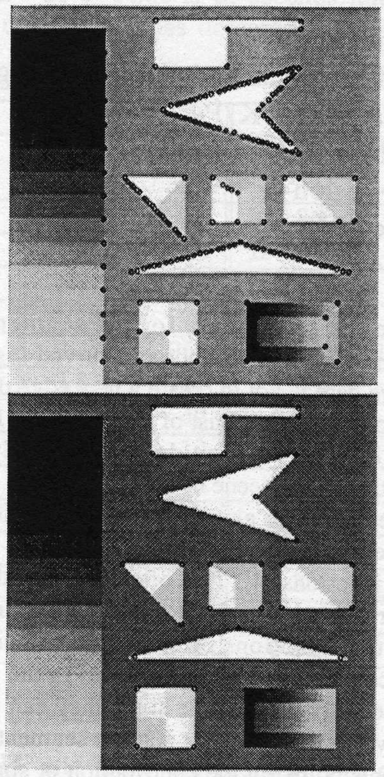
As will be seen in Section 2, several approaches were proposed to detect junction points in an intensity image. Unfortunately, they all suffer from a significant proportion of false detections. An improvement to this would result from using an additional stage of classification based on criteria describing the junction. In effect, such criteria would provide further information and allow a better selection of real junctions. This would in turn give a primary description of the underlying object structures. Besides, robust junctions can be used as starting points for an edge linking procedure since they provide, for instance, the initial search directions. Hence, a complementary detection of edge and junction points is likely to lead to a more significant reconstruction of edge point contours.

In this paper, we aim to show one way junction points can be validated and classified starting from unlinked edge points map. The paper is organized as follows: in Section 2, a brief review of existing junction detection and classification methods is presented. Section 3 introduces the proposed method for junction validation and classification. A region of interest is to be associated to each junction candidate. Then, Section 4 shows experimental results for synthetic and real

Fig. 1 Example of junction detection by Rosenthaler. (a)Threshold: 5%, (b) Threshold: 60%.

(a)

(b)



images with discussion of the performances. Finally, section 5 gives a brief summary of our approach and perspectives of research.

2. Review

Many approaches have been developed for detecting junctions from an intensity image. Among the most recent, Smith and Brady use region intensity statistics to provide a rapid detection of junctions [12]. This approach suffers from bad junction localization and can fail to detect some real junctions. Rosenthaler proposes a junction detection scheme based on the analysis of oriented energy representations [11]. This method, according to our knowledge, is the best one in regard to the proportion and completeness of good detections. However, it can present some limitations in the thresholding step which is based only on an amplitude criteria. An example of this is illustrated in Fig. 1. When the threshold is low, all junction points are detected but many false points too. To eliminate false points, the threshold value is increased leading to the disappearance of "T" junctions.

As stated before, a classification step applied on the junction candidates improves the description of the underlying structures in two manners. At first, it allows to validate junctions with strict criteria such as the number of branches associated and the angles separating them. Secondly, the information acquired from this step helps the edge linking process.

There exists a number of methods which try to classify junctions from the intensity image. The most recent ones use parametric junction models to which are fitted intensity data [10][3]. The main drawback associated with these methods is the need to model each type of junction. This can lead to a somewhat complex system depending on how many models are to be considered. Besides, the multiple parameters required to precisely model the acquisition system add to the difficulty.

These arguments clearly point out the need for a junction classification method without specific models. A method based on a grouping procedure of edge points according to perceptual grouping criteria could achieve this goal. We have developed such a bottom-up approach which starts from the unthresholded edge map to extract significant junctions as well as their describing characteristics. The following sections present the proposed algorithm.

3. Proposed Approach

This section describes the tree main parts of the proposed approach. The first point discusses the method providing initial edge point data and junction candidates. Then, the junction characterization process is described. It is based on a binary splitting procedure applied on vectors associated to edge points. This process provides in output the junction branches. These are used to refine the junction position. Fitted lines finally allow junction validation and classification.

3.1 Initial Data

The choice of an adequate edge detector is of primary concern. The reason is the whole process of junction classification relies on the analysis of detected edge points. For instance, since we want to study the arrangement of edge points in the neighborhood of junctions, we expect all these points, without gaps, as well as a precise estimation of their orientation. Nonetheless, many edge detectors produce gaps at junctions if the constituent edges have large differences in contrast. This is not the case of Heitger's detector which yields a good continuity of features near a junction [7]. Moreover, a good detection and a precise estimation of gradient orientation is provided by the combination of responses due to filters at n different orientations[1]. For all these reasons, we selected Heitger's edge detector as part of our approach.

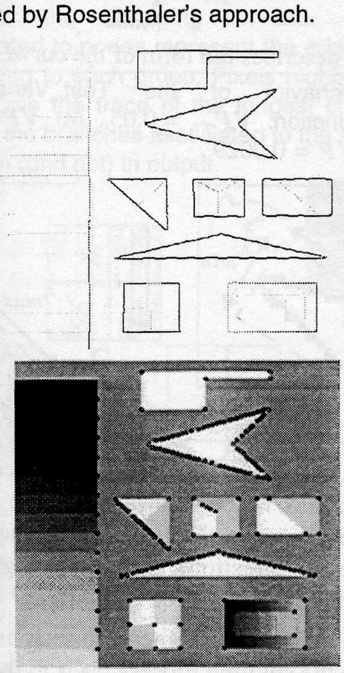
In order to focus on regions of interest automatically, without the help of a human operator, a

1. We have chosen 6 orientations as Heitger did.

Fig. 2 Input data for the junction classification process. (a) edge map provided by Heitger's detector. Each pixel intensity displays the amplitude obtained after combination of responses over orientation. (b) Junctions detected by Rosenthaler's approach.

(a)

(b)



junction detection step need first to be applied. We have chosen Rosenthaler's method [11] mainly because of its concordance with Heitger's edge detector. For instance, these two methods use the same intermediate representation so they can be applied simultaneously. Fig. 2 displays initial data for a standard test pattern [12]. In edge map, each pixel intensity displays the amplitude of the first Fourier harmonic of the n-tuple of responses over orientation. As can be observed, many false points are marked among the detected junctions. We voluntarily chose a low threshold in order to retain all real junctions.

In summary, we now dispose of potential junction points defining regions of interest and an edge map. It is important to notice that non maximum suppression is activated but no thresholding is applied on the edge points. Besides, each of these points is associated with a vector of n elements which describes the responses to the n oriented filters.

3.2 Binary Splitting

We aim at finding locally straight sharp lines which describe each of the branches associated to a junction.Therefore, we need to form edge point groupings corresponding to junction branches. Each formed edge point group must present an edge point configuration which looks like the desired line configuration.

The basic idea of our algorithm is to cluster edge points in groups according to their similarity of orientation and amplitude in each oriented channel. Each vector of n elements associated to an edge point e has the form:

$$\underline{v}_e = [a_1 \dots a_i \dots a_n]^T \quad (1)$$

where a_i is the response of the convolution with the i[th] oriented mask.

As a starting assumption, we consider that all vectors $\underline{v}_e$ of edge points included in a region of interest belong to the same group G.

The methodology we adopt is a binary splitting procedure which divides a group G into two subgroups G_1 and G_2 according to the direction of maximal deviation. This is equivalent to partition edge points using the direction of the eigenvector associated with the higher eigenvalue of the covariance matrix. A similar principle is used in vector quantization methods to determine the number and the initial values of quantized elements. Gray[6] has pointed out the advantages of using vectors instead of scalar values for that matter. Likewise, the vectors used here express strong relations for a particular edge point between orientation and amplitude.

For a given group G, the covariance matrix R is defined as:

$$R = \sum_{\underline{v}_e \in G} (\underline{v}_e - \underline{\mu})(\underline{v}_e - \underline{\mu})^T \quad (2)$$

where $\underline{\mu}$ is the centroid of the group G defined by:

$$\underline{\mu} = \frac{1}{N} \cdot \sum_{\underline{v}_e \in G} \underline{v}_e \quad (3)$$

From the above matrix, one can compute the associated eigenvectors and eigenvalues. Let us define as λ_k and $\underline{\phi}_k$ $(k = 1 \dots n)$, the k[th] eigenvalue and eigenvector of R, respectively. We search $\underline{\phi}_{max}$ such as:

$$\underline{\phi}_{max} = \max_{\underline{\phi}_k}(\underline{\phi}_k^T \cdot R \cdot \underline{\phi}_k) \quad (4)$$

$\underline{\phi}_{max}$ allows us to split the group G into two subgroups G_1 and G_2:

$$G_1 = \left\{ \underline{v}_e \in G, \underline{v}_e^T \underline{\phi}_{max} \le \underline{\mu}^T \underline{\phi}_{max} \right\} \quad (5)$$

$$G_2 = \left\{ \underline{v}_e \in G, \underline{v}_e^T \underline{\phi}_{max} > \underline{\mu}^T \underline{\phi}_{max} \right\} \quad (6)$$

The algorithm splits the groups iteratively until the variation of components in each group is below a fixed threshold. The resulting groups contain the edge points which are considered to belong to the same underlying structure.

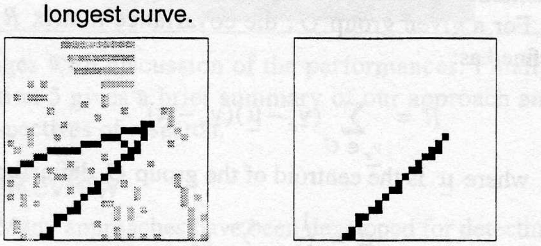
The main difficulty relative to this method is to define the stopping rules of the iterative process. Our tests showed that the variation measure was not significant enough to be used as a stopping criterion. How can we decide that data of a group are homogeneous enough and do not need to be separated anymore? As was stated before, we search elongated lines one pixel wide. This leads us to privilege groups of elements which respect criteria of connectivity and topology.

We have defined two stopping functions which take into account these criteria. They are:

- **Test_Connectivity function:**
  This function is designed to search the largest number nb of connected edge points in a group of N elements. It defines the confidence we can have in the presence of a unique curve in the group by using a threshold C_{Th}. The function can be written mathematically as:

$$C = 1 - \frac{nb}{N} \tag{7}$$

Fig. 3 Behavior of the Test_Connectivity function. $C_{Th} = 0.2$. (a) $C = 0.698$, (b) $C = 0$. Pixels highlighted in black give the trace of the longest curve.



(a) (b)

Fig. 3 shows the behavior of the function. When many edge points belonging to one group are isolated, C is over C_{Th}. On the other hand, a value of C nearby zero gives high confidence edge points in the group belong to a unique curve.

- **Test_Variance_Position function:**
  In this function, only edge points belonging to the longest curve are considered. The position of an edge point with respect to others is studied in order to verify the curve coherence considering this can easily be obtained by eigenvalues associated to the covariance matrix of edge point position vectors $p_e = (x,y)$. The position covariance matrix R_p is written as:

$$R_p = \sum_{p_e \in G} (\underline{p}_e - \underline{\mu}_p)(\underline{p}_e - \underline{\mu}_p)^T \tag{8}$$

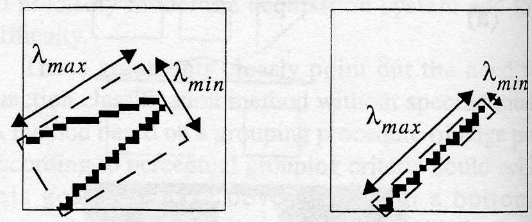
where $\underline{\mu}_p$ is the mean position vector.

The two eigenvalues λ_{min} and λ_{max} can be computed and we define the measure:

$$VP = \frac{\lambda_{min}}{\lambda_{max}} \tag{9}$$

which describes the form of the curve.

Fig. 4 Behavior of the Test_Variance_Position function. $VP_{Th} = 0.05$. (a) $VP = 0.347$, (b) $VP = 0.0029$.



(a) (b)

As before, the measure VP is compared to a threshold which allows to decide whether edge points forming the longest curve are scattered in several directions or are aligned in one predominant direction (Fig. 4).

The application of these functions as stopping criteria in the splitting process leads to the formation of an unbalanced binary tree for each region of interest (Fig. 5). This tree allows to point out all coherent structures which can be associated to junction branches.

3.3 Junction Validation and Classification

Once all lines passing through the region of interest have been detected, we need to decide if a junction is really present. We now describe how this can be achieved.

The junction position is refined and validated simultaneously. This is realized by searching the position which minimizes the distance to local tangents of edge points. This approach is a modified version of Förstner's algorithm in which the participating points are selected on the lines and gradient amplitude is not considered [5]. This reasoning has two main advantages over Förstner's method. At first, it considers edge points even if they lie on a region of small contrast. Secondly, the influence of non edge points is eliminated. The algorithm provides in output the final junction position with subpixel precision. A form factor is also obtained which allows to validate the junction.

Finally, curve fitting is realized in order to classify the junction element. Up to now, we limit our fitting to straight lines but a generalization to curved segments is planned. A straight line L is defined as:

$$ax + by + c = 0 \tag{10}$$

Fig. 5 Example of an unbalanced tree obtained by the binary splitting process. (a) Initial region of interest with the junction detected by Rosenthaler. (b) Binary tree created. The shadowed nodes (G_2, G_{12}) are those which respect the stopping criteria. Images associated to nodes represent the edge points belonging to each group. Pixels highlighted in black give the trace of the longest curve. (c) shows the branches associated to the potential junction (grid dot) in output.

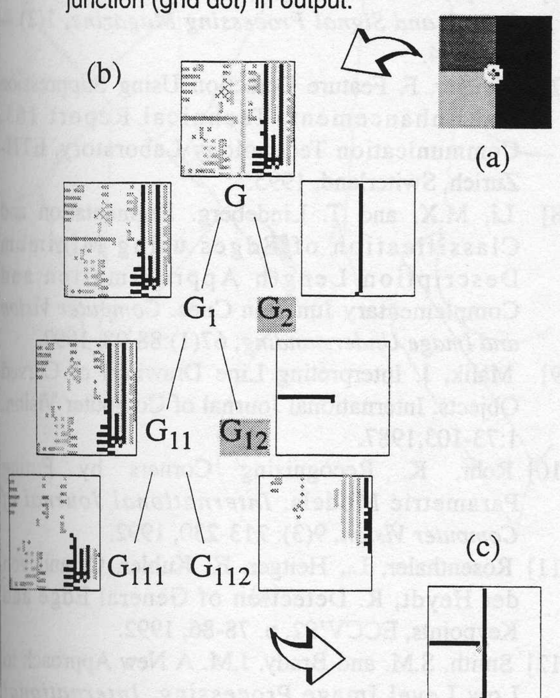
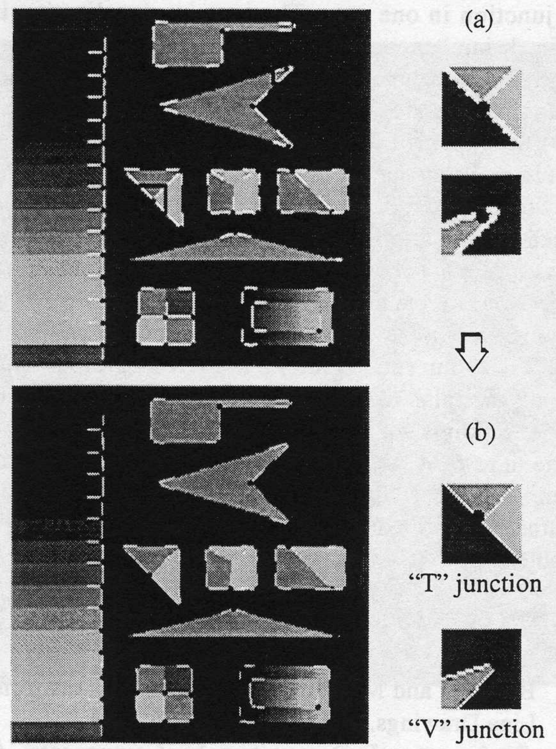


Fig. 6 Example of junction classification. (a) presents the output after the binary splitting process. (b) displays junction localization refinements and fitted straight lines associated to junctions. The two regions of interest selected show the detection of a "T" junction and a "V" junction.



The fitting parameters of L are easily obtained from the covariance matrix of position vectors calculated in the Test_Variance_Position function. For instance, the eigenvector associated to the largest eigenvalue is directly related to the straight line parameters a and b, so we can easily get the fitting estimates (Fig. 6).

Finally, the junction classification may be obtained by comparing the configuration of the fitted lines to the general catalog defined by Malik [9].

4. Results

Fig. 7 and Fig. 8 show the initial data and the results obtained by applying the junction classification process. They illustrate the type of results which can be obtained for a synthetic image and a real image respectively. Notice how false junction points detected by Rosenthaler have been eliminated with our method while keeping and validating almost all real junctions for both images. The results are promising since almost all significant junctions have been validated and classified in an unified way without the need of any specific model. Junctions of different types (such as occlusion, arrow, Y junctions) have been detected simultaneously and all false junction points have been eliminated.

Besides, one can imagine the help junction points and their associated branches could bring to the line formation process. For instance, each branch provides a good indication of the form of the underlying object curve. It shows the initial direction of search for connecting edge points. Moreover, branches detected by the binary splitting process allow to extract edge points nearby a junction which are generally lost by a classical thresholding step. This is illustrated for example in the real image (Fig. 8) on the region pointed by the black arrow. One can notice the recovery of cone edge points by the binary splitting process despite their low amplitude.

5. Conclusion

We have proposed a new method of junction validation and classification using edge points and regions of interest. The originality of our approach relies on the absence of parametric models to classify junctions.

We have described how the binary splitting procedure allows to naturally cluster edge points according to perceptual grouping criteria such as similarity, proximity or simplicity. Moreover, the classification is done in an unified way for all junction types whereas classical approaches classify each type of junction in one pass. The junction localization is refined simultaneously with subpixel accuracy. In most cases, the new junction position appears more precise. A comparison study is to be done in order to precisely evaluate in which way the position refinement method can improve the junction localization.

Some interference problems can still be encountered when two junctions are present in the same region of interest. Future research will consider a scale factor in order to adapt the interest window size according to the feature size. Besides, the generalization of the fitting process to curved segments is under development. Finally, we plan to use the obtained junction points as starting points for the contour lines process. This is done in respect with the philosophy of considering the combination of edge and junction points as a richer source of information for real-world object recognition problems [8].

6. References

[1] Bergevin, R. Generic Object Recognition: Building and Matching Coarse Descriptions from Line Drawings. PAMI, 15(1): 19-36,1993.

[2] Biedermann, I. Recognition by Components: A Theory of Human Image Understanding. Psychological Review, 94(2):115-147,1987.

[3] Blazcka, T. and R. Deriche. Recovering and Characterizing Image Features using an Efficient Model Based Approach. Research report 2422, INRIA, november 1994.

[4] Fleck, M.M. Some Defects in Finite-Difference Edge Finders. *PAMI*, 14(3): 337-345,1992.

[5] Förstner, W. and E. Gulch. A Fast Operator for Detection and Precise Location of Distinct Points, Corners and Centres of Circular Features. *In Intercomission conf. on Fast Proceeding of Photogrammetric Data*, Interlaken, p. 281-305, June 1987.

[6] Gray, R. M. Vector Quantization. *IEEE Acoustics, Speech and Signal Processing Magazine*, 1(2):4-29, 1984.

[7] Heitger, F. Feature Detection Using Suppression and Enhancement. Technical Report 163, Communication Technology Laboratory, ETH-Zurich, Switerland, 1995.

[8] Li, M.X. and T. Lindeberg. Segmentation and Classification of Edges using Minimum Description Length Approximation and Complementary Junction Cues. *Computer Vision and Image Understanding*, 67(1):88-98, 1997.

[9] Malik, J. Interpreting Line Drawings of Curved Objects. International Journal of Computer Vision, 1:73-103,1987.

[10] Rohr, K. Recognizing Corners by Fitting Parametric Models. *International Journal of Computer Vision*, 9(3): 213-230, 1992.

[11] Rosenthaler, L., Heitger, F., Kubler, O. and von der Heydt, R. Detection of General Edge and Keypoints, ECCV'92, p. 78-86, 1992.

[12] Smith, S.M. and Brady, J.M. A New Approach to Low Level Image Processing. *International Journal of Computer Vision*, 23(1):45-78, 1997.

Fig. 7 Results obtained on a synthetic image. Input data: (a) edge map by Heitger's approach, (b) potential junctions detected by Rosenthaler. (c) Validated junctions and their associated branches after the binary splitting process. (d) Final output after the junction position refinement and the fitting process on each branch.

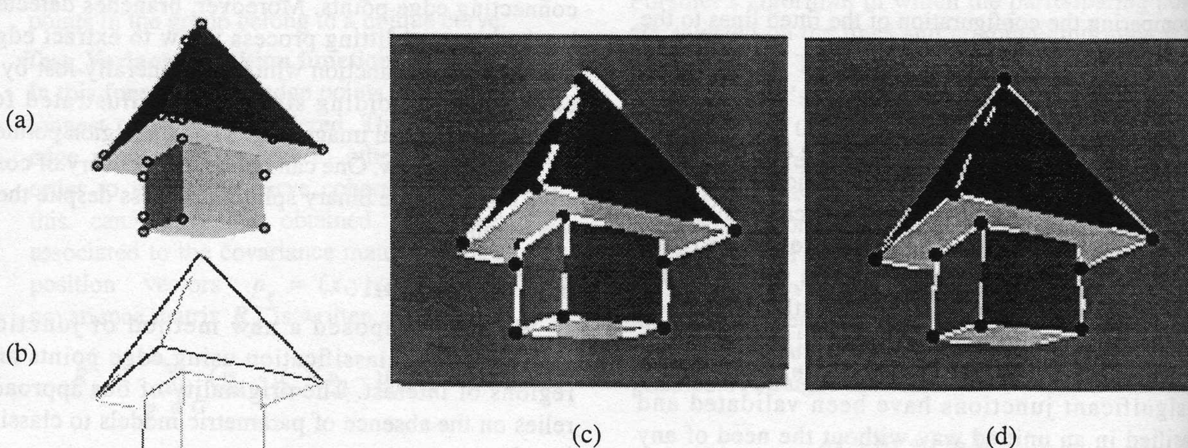


Fig. 8 Results obtained on a real image. Input data: (a) edge map by Heitger's approach; (b) potential junctions detected by Rosenthaler; (c) validated junctions and their associated branches detected by the binary splitting process; (d) final output after the junction position refinements and the fitting process on each branch. Observe the occlusion junction (indicated by the black arrow) between the pyramid and the cone and edge points of the corresponding region of interest. The upper edge of the cone seems difficult to recover if we look at the given edge points. Notice how the associated branches of this junction may help the process of contour line extraction.