Perspective Rectangle Detection

David Shaw and Nick Barnes

Autonomous Systems and Sensing Technologies Programme
National ICT Australia
Research School of Information Sciences and Engineering
Australian National University
`david.shaw@rsise.anu.edu.au` `nick.barnes@nicta.com.au`

Abstract. This paper describes a new detector for finding perspective rectangle structural features that runs in real-time. Given the vanishing points within an image, the algorithm recovers the edge points that are aligned along the vanishing lines. We then efficiently recover the intersections of pairs of lines corresponding to different vanishing points. The detector has been designed for robot visual mapping, and we present the application of this detector to real-time stereo matching and reconstruction over a corridor sequence for this goal.

1 Introduction

Using vision for Simultaneous Localisation and Mapping (SLAM) is a key topic of research in mobile robotics. While there has been present success with data received from laser rangefinders[1], cameras provide a cheaper and more information rich sensor for determining the nature of the environment. However the denser nature of visual information makes it harder to extract the discrete landmark information to build maps than with rangefinders. With the use of appropriate visual feature detectors, information can be extracted for practical use for robotic applications such as SLAM. Some of the more popular feature detectors used in general vision applications are the SIFT descriptor[2, 3], the Harris point detector[4] and variants[5, 6], for use in domains such as recognising locations[7], objects[2, 8], matching for image retrieval[5], and for our target domain of robotic navigation[9].

For built environments, such as indoor scenes, the *a priori* knowledge about the geometry of the environmental structure can be used as the basis of a higher-order feature detector. In many such domains, the environment consists of predominant rectilinear structure. With rectangles making up much of the environmental geometry, it is prudent to use this as the basis of a feature detector. Since most rectangles in the environment will not be aligned parallel to the camera, standard rectangle detectors such as those used for building extraction from aerial images[10] are not applicable as rectangular features will appear as trapeziums to a robot camera.

In this paper, we introduce the perspective rectangle detector for finding rectangles projected onto the image plane. The detector locates the likely 2D projection of rectangles in the environment by locating quadrilaterals with sides aligning with detected vanishing lines within the image. While such features can be detected with methods such as a generalised Hough transform[11] or via perceptual grouping[12], the order of complexity of such approaches make them unsuitable for real-time applications.

The perspective rectangle detector is designed to provide accurate feature detection within the real-time operational constraint. We provide the outline of the real-time algorithm behind the detector, and discuss the result of the detector being run in sample domains.

2 Algorithm Description

Finding all possible quadrilateral features within a scene is not a feasible approach to constructing a real-time detector, as calculating the full probability density function is intractable. Neither is a Hough based approach feasible, as the problem is eight dimensional (two per vertex), and thus too computationally expensive.

Given that for our application we require only to find the strong features within the environment, we can make a significant reduction in complexity by narrowing the domain to perspective rectangular features: those aligned with vanishing lines within the scene. This geometric assumption presumes that the environment consists of rectangles on planar walls and ceilings, but this is a common occurrence in indoor environments. If we are searching for the perspective rectangles between two given sets of vanishing lines, the problem is reduced to four dimensions, which makes the problem more manageable, but still too computationally expensive for a Hough-based approach. To solve this problem, we have devised an algorithm for finding such perspective rectangular features, an outline of which is provided below:

1. *Vanishing Point Detection:* Estimate the position and type of vanishing points and lines within the image.
2. *Perspective Oriented Edge Detection:* Determine the directional components of the gradient aligned with the vanishing points.
3. *Line Segment Detection:* Estimate the line segments that are potential quadrilateral sides based on the perspective oriented edges.
4. *Quadrilateral Detection:* Determine the quadrilaterals from the intersection of detected line segments.

2.1 Vanishing Point Detection

Vanishing points and the corresponding sets of vanishing lines provide key information about the predominant geometric structure of the environment. While there are many vanishing point detectors available[13–16], due to the need for the detector to run real-time in indoor environments, we are using our own vanishing point detector, based on the distance of candidate vanishing points from lines within the image. Other vanishing point detectors can be substituted if they are real-time and suitable for the task and environment.

2.2 Perspective Oriented Edge Detection

Once the sets of vanishing lines have been determined, the edges that correlate with these sets need to be found. Edge detection is simply done by applying a convolution mask such as the Sobel edge detector. Finding the associated directional edge vector field (which we denote $\mathbf{E}$) culls many points that are unlikely to be part of a vanishing line. The edge vector field in the direction of the vanishing lines can be found with the dot product $\mathbf{D} = \mathbf{E} \cdot \mathbf{V}$; where $\mathbf{V}$ is the vector field consisting of unit vectors in the direction of the vanishing lines. $|\mathbf{D}|$ represents the distance in orientation of the edge point to the vanishing line, and so gives the likelihood of the vanishing line at that point. Since the Sobel edge detector blurs edge features this also provides an approximate distance function over a window of a few pixels.

2.3 Line Segment Detection

To find the vanishing line segments, edge vectors for a particular vanishing point are scanned with a scan-line algorithm. The blur provided by the Sobel edge detector makes candidate lines more readily detected by this approach. The operation of the algorithm is slightly different depending on the type of input vanishing lines. For parallel lines, the algorithm scans through every possible discrete parallel vanishing line (aligned with θ); for radial, it scans through each ray (radiating out from the vanishing point) through a fixed angle step (presently defined to half a degree).

To deal with noise within the image, $|\mathbf{D}|$ is thresholded at a pre-determined value μ_p, with all values where $|\mathbf{D}| \geq \mu_p$ being considered viable line points. The algorithm also allows a fixed number (μ_g) of consecutive "misses" when scanning through a ray, with each run of line points with no gaps greater than length μ_g considered line segments. This approach may prune some weak candidate lines, in practice though it finds enough line segments to be suitable for applications.

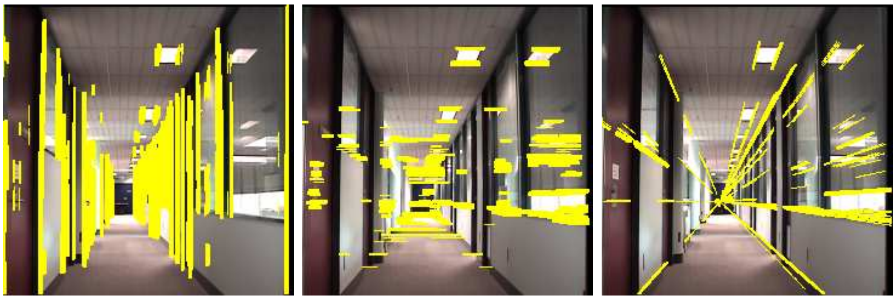

Fig. 1. The line segments found. *left image*: vertical line segments; *centre image*: horizontal line segments; *right image*: line segments oriented to the vanishing point.

2.4 Quadrilateral Detection

The quadrilateral detection algorithm runs on a pair of sets of vanishing line segments defined by the two sets of vanishing lines that their sides are aligned with, which we will denote S_a and S_b. A quadrilateral feature is present if, for pairs of line segments $(a_1, a_2) \in S_a$ and $(b_1, b_2) \in S_b$, both of a_1 and a_2 intersect both of b_1 and b_2 from the viable line segments detected previously.

Coordinate Transformation: Each set S_i (of S_a and S_b) is translated to its own coordinate system based on the type of vanishing point (scale is unimportant); radial: based on angle orientation from the vanishing point; parallel: based on an offset from the zero line, defined as the line that passes through the image origin. A line segment is represented as a triple using the coordinate systems of both sets: a segment in S_a is defined $(\alpha, \beta_1, \beta_2)$, where α is the reference to the vanishing line in S_a, and β_1 and β_2 are the end points of the line segment expressed in terms of S_b. This representation aids in finding the intersections between the sets.

Segment Quantisation: The line segments of the transformed S_i are discretised into a 2D array of cells. A line segment is referenced in every cell that it crosses, so that only those segments which share a cell reference need to be compared for possible intersections.

Intersection Pairing: A list of all intersections between line segments is constructed using the segment quantisation for speed. To find the quadrilaterals, all intersecting pairs of line segments from S_a with pairs from S_b are found in the following manner:

- For every line segment $a \in S_a$
  - For each $b \in S_b$ that intersects a, count the $a_b \in S_a$ intersect these b (except for $a_b = a$)
  - For each a_b that was counted at least twice (i.e. there are at least two $b \in S_b$ that a_b intersects), go through each $b \in S_b$ and record a list of which b intersect a_b
  - For the list of b every combination of pairs of line segments will intersect both a and a_b, and thus these four line segments will be a valid quadrilateral

A valuation of the likelihood strength of the quadrilateral feature is approximated by summing the strength of its constituent line segments.

Trimming Multiple Features: Due to the nature of the vector edge field, it is possible for multiple line segments to be detected for a single true edge, resulting in multiple quadrilaterals being found for a single feature. Thus as a final step those detected quadrilaterals that have all four of their corner vertices in the same small neighbourhood (within four pixels) are compared, and only that with the greatest likelihood strength is kept.

Fig. 2. Recovered perspective rectangular features. *left:* from vertical line segments and line segments leading to the vanishing point. *centre:* from horizontal line segments and line segments leading to the vanishing point. *right:* from horizontal and vertical line segments.

3 Applications and Results

The prototype version of the algorithm (running in Matlab[1]) performs matching between stereo images at an average of 2.54 seconds (worst case under 4 seconds), including vanishing point detection, for test video sequences of 320×240 resolution. For the domain of robotic SLAM, the prototype implementation is running at speeds suitable for real-time performance. In order to best view the results of the detector, a video of matched features has been provided as an attachment. This video of a corridor loop sequence shows the type of features found by the detector in a real-world environment.

Note that for the results shown the two most predominant sets of infinite vanishing points are vertical and horizontal lines, due to the design of our building, so these are used by default as two of three sets of vanishing lines (with the vanishing point detector providing the third). This can easily be extended with an additional detection stage which utilises an additional step to find the predominant sets of parallel lines within an image.

3.1 Sample Perspective Rectangle Features

The test was run over a sequence of 1000 calibrated stereo images taken by our robot[2]. Figure 3 shows perspective quadrilateral features in a number of stereo pairs taken from this sequence (images were processed at 320×240). The strongest vanishing point detected is shown in the images as a green 'X'. All three sets of features are shown on the same frame for printing space reasons. Stronger features are shown thicker and in yellow, with weaker features in orange, weakest in red. As can be seen, much of the structure of the environment is found as features corresponding mostly to the walls and ceiling of the corridor. Average processing time is 1.12 seconds.[3]

3.2 Matching and 3D Reconstruction

A key target domain for the perspective rectangle feature detector is real-time matching of stereo features for scene reconstruction or mapping. Since there is much implicit structural information contained within a single feature it is possible to gain a lot of information about a built environment.

The features found (figure 3) are matched with a simple metric based on vertex position: two features are considered to be possible matches if all four vertices are within a vertical difference of three pixels and a horizontal difference of 20 pixels. If there is a unique match between two quadrilaterals then it is considered a viable matched feature.

[1] At time of publication the detector is being converted into C for fast performance.
[2] Nomadics Technology XR4000 Mobile Robot
[3] The complete video sequence of the perspective rectangle features found in the stereo images is provided at time of publication at `http://users.rsise.anu.edu.au/~davids/video/prd-video.zip`.

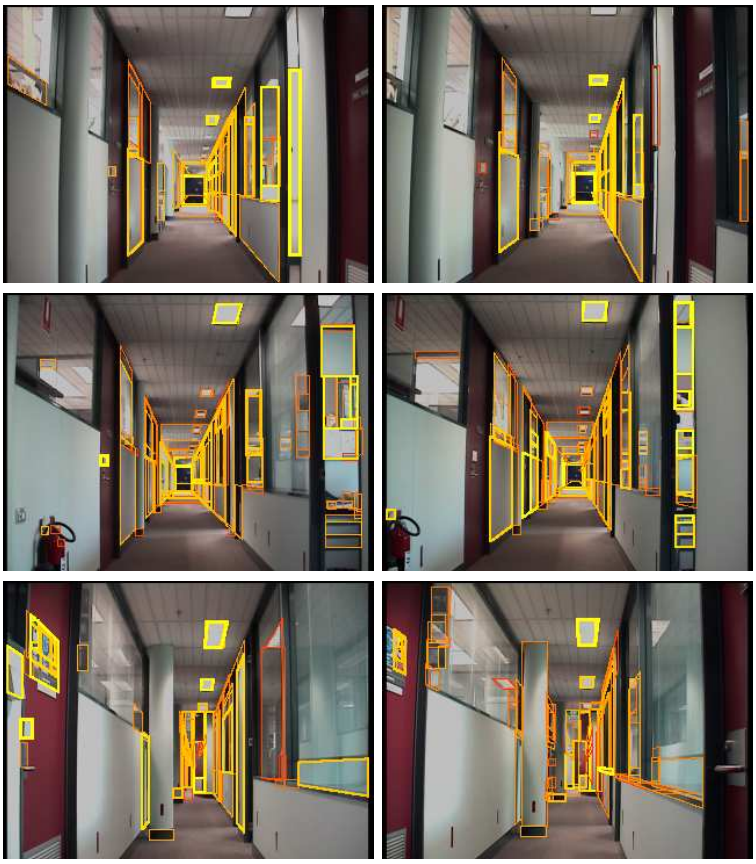

Fig. 3. Perspective rectangle features in a corridor sequences.

The same image sequence is used as in section 3.1. Figure 4 shows sample images with their matched features. The simulated output of a virtual camera (located above and to the left of the real cameras) is provided to show where the position of these features are from a separate angle. As can be seen from figure 4, even with a simple matching metric and with persistent tracking of the features throughout concurrent frames in the sequence, there is significant structural detail being obtained from the raw perspective rectangle feature detector alone.

4 Conclusion

Features based on the structure of built environments are useful for a number of domains, such as scene reconstruction and SLAM. This paper has presented a new perspective rectangle feature detector for finding structure based features in built environments from images that runs in real-time on real world data. Based on vanishing point information, a perspective rectangle feature contains implicit

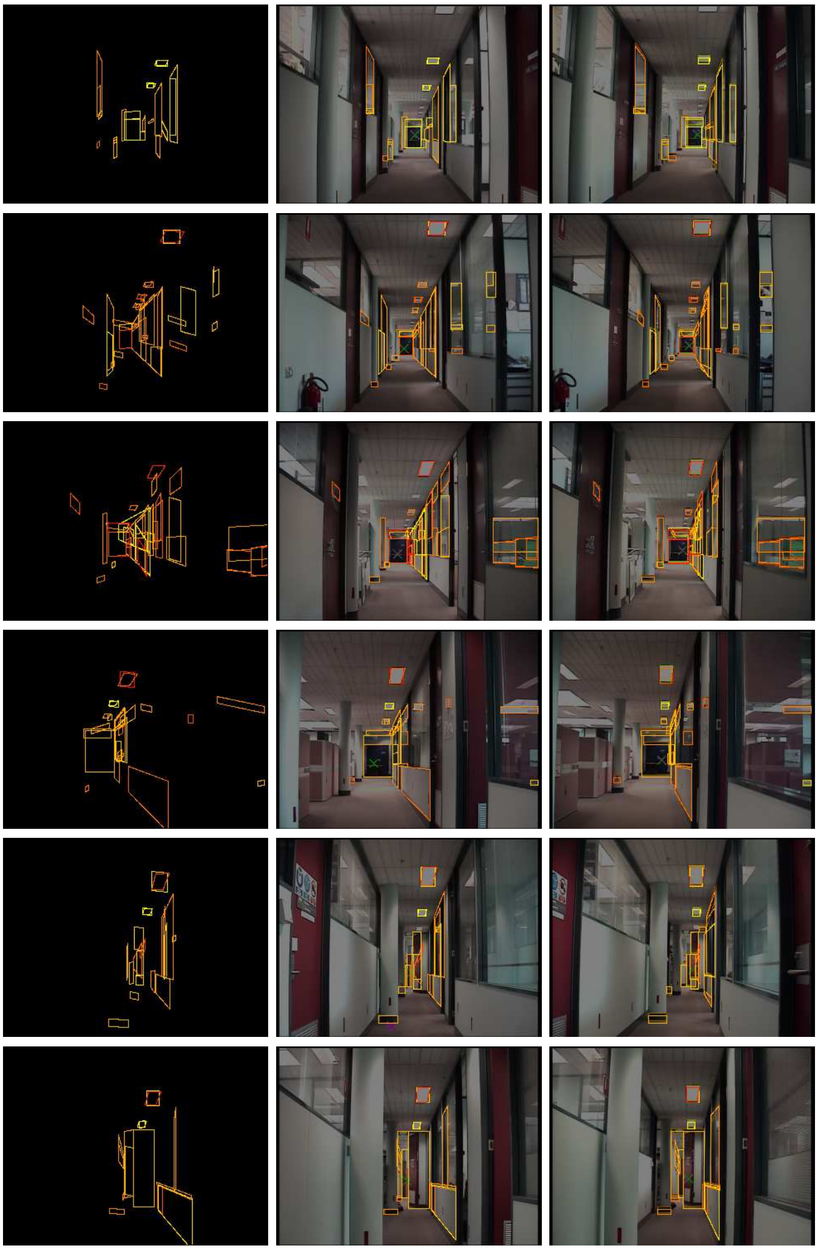

Fig. 4. Matching perspective rectangle features in a corridor sequence, samples taken from the 1000 frames in the video sequence. Centre and right pictures show matched features in the image. Left pictures show 3D rectangle features viewed from a 'virtual camera'

structural information that is of benefit for robotic applications. We show that the detector can find features that, even with a simple matching metric, provides significant structural information about a scene.

References

1. Leonard, J.J., Durrant-Whyte, H.F.: Simultaneous map building and localization for an autonomous mobile robot. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). (1991) 1442–1447
2. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proceedings of the International Conference on Computer Vision, Corfu (1999) 1150–1157
3. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60** (2004) 91–110
4. Harris, C., Stephens, M.: A combined corner and edge detector. In: Alvey Vision Conference. (1988)
5. Mikolajczyk, K., Schmid, C.: Indexing based on scale invariant interest points. In: Proceedings of the 8th International Conference on Computer Vision, Vancouver, Canada (2001) 525–531
6. Mikolajczyk, K., Schmid, C.: An affine invariant interest point detector. In: European Conference on Computer Vision, Copenhagen (2002) 128–142
7. Silpa-Anan, C., Hartley, R.: Localisation using an image-map. In: Australasian Conference on Robotics and Automation. (2004)
8. Sivic, J., Shaffalitzky, F., Zisserman, A.: Object level grouping for video shots. In: European Conference on Computer Vision, Copenhagen (2002) 85–98
9. Se, S., Lowe, D., Little, J.: Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. International Journal of Robotics Research **21** (2002) 735–758
10. Vinson, S., Cohen, L.D.: Multiple rectangle model for buildings segmentation and 3d scene reconstruction. In: Proceedings of the 16th IAPR International Conference on Pattern Recgonition. (2002)
11. Ballard, D.H.: Generalizing the hough transform to detect arbitrary shapes. Pattern Recognition **13** (1981) 111–122
12. Guy, G., Medioni, G.: Inferring global perceptual contours from local features. Int. J. Comput. Vision **20** (1996) 113–133
13. Barnard, S.T.: Interpreting perspective images. Artificial Intelligence (1983)
14. Collins, R.T., Weiss, R.S.: Vanishing point calculation as a statistical inference on the unit sphere. In: International Conference on Computer Vision. (1990) 400–403
15. Cantoni, V., Lombardi, L., Porta, M., Sicard, N.: Vanishing point detection: Representation analysis and new approaches. In: Proceedings of the 11th International Conference on Image Analysis and Processing. (2001) 90–94
16. Rother, C.: A new approach to vanishing point detection in architectural environments. In: Architectural Evnironments British Machine Vision Conference (BMVC), Bristol, GB (2000) 382–391