The Gixel Array Descriptor (GAD) for Multi-Modal Image Matching

Guan Pang  
University of Southern California  
gpang@usc.edu

Ulrich Neumann  
University of Southern California  
umeumann@graphics.usc.edu

Abstract

Feature description and matching is a fundamental problem for many computer vision applications. However, most existing descriptors only work well on images of a single modality with similar texture. This paper presents a novel basic descriptor unit called a Gixel, which uses an additive scoring method to sample surrounding edge information. Several Gixels in a circular array create a powerful descriptor called the Gixel Array Descriptor (GAD), excelling in multi-modal image matching, especially when one of the images is edge-dominant with little texture. Experiments demonstrate the superiority of GAD on multi-modal matching, while maintaining a performance comparable to several state-of-the-art descriptors on single modality matching.

1. Introduction

Finding precise correspondences between image pairs is a fundamental task in many computer vision applications. A common solution is extracting dominant features to describe a complex scene, and comparing similarity between feature descriptors to determine the best correspondence. However, most descriptors only perform well with images of the same modality. Successful matching depends on similar distributions of texture, intensity or gradient.

Multi-modal images usually exhibit different patterns of gradient and intensity information, making it hard for existing descriptors to find correspondences. Examples include matching images from SAR, visible light, infrared devices, Lidar sensors, etc. A special case in multi-modal matching is to match a traditional optical image to an edge-dominant image with little texture, such as matching optical images to 3D wireframes for model texturing, matching normal images to pictorial images for image retrieval, matching aerial images to maps for urban localization, etc. These tasks are even harder for existing descriptors since they often don’t have any texture distribution information.

To match multi-modal images, especially edge-dominant images, we present a novel gradient-based descriptor unit called a “Gixel”, abbreviated from “Gradient Pixel”. A Gixel is a sample point for the gradient information in a small local region. It serves as a basic unit of the complete descriptor - the Gixel Array Descriptor (GAD), which consists of several Gixels in a circular array. Each Gixel is used to sample and score gradients in its neighborhood. The scores are normalized to represent relative gradient distribution between the Gixel regions, forming a complete descriptor vector. Owing to the circular array, the descriptor is easily adjusted to any orientation and size, allowing invariance in both rotation and scale.

It is our observation that line features are the most important and reliable features in multi-modal matching applications. A key property of a Gixel is that it samples the information of a long connected line with much the same result as it samples a series of short dashed line segments. This is
achieved by an additive scoring method. The score obtained with small line segments of similar orientation and distance to the Gixel is similar to the score obtained from a long connected line of the same orientation and distance. In multi-modal data, due to different sensor characteristics, line features may appear with gaps or zig-zag segments. Gixel’s additive scoring method samples broken or solid lines with similar scores. This is essential for multi-modal matching. Figure 1 shows an example matching result of the Gixel-based descriptor, between an ordinary image and a “pencil drawing” stylized image.

Our main contributions include:

- We observe that line features are the most important and reliable features in many multi-modal matching applications.
- We propose the Gixel Array Descriptor (GAD) to extract line features, and thus perform robustly for matching multi-modal images, especially on edge-dominant images with little texture.
- We demonstrate that the GAD can achieve good matching performance in various multi-modal applications, while maintaining a performance comparable to several state-of-the-art descriptors on single modality matching.

2. Related Work

The existing keypoint descriptors can be roughly classified into two categories: The ones based on intensity or gradient distribution, and the ones based on pixel comparison or ordering.

The distribution-based descriptors are more traditional, which can be dated back to the works of Zabih and Woodfill [3], Johnson and Hebert [4] and Belongie et al. [5]. An important representative descriptor in this category is SIFT (Scale-Invariant Feature Transform) [6], which coded orientation and location information of gradients in the descriptor region into a histogram, weighted by magnitude. Ke and Sukthankar [7] reduced the complexity of SIFT with PCA, leading to PCA-SIFT which is faster but also less distinctive than SIFT. Mikolajczyk and Schmid [1] proved the great performance of SIFT, and proposed a new descriptor based on SIFT by changing its location grid and reducing redundancy through PCA. The resulting GLOH is more distinctive than SIFT, but at the same time more expensive to compute. Bay et al. [2] proposed SURF (Speeded-Up Robust Features), using integral images to reduce the computation time, while still retaining the same gradient distribution histograms of SIFT. SURF [2] has proved to be one of the best descriptors in various circumstances.

Recently, comparison-based descriptors are attracting attention. These descriptors use relative comparison or ordering result of pixel intensities rather than the original intensities, resulting in much more efficient descriptor computation, while still maintaining performance competitive to distribution-based descriptors such as SIFT or SURF. Tang et al. [10] proposed a 2D histogram in the intensity ordering and spatial sub-division spaces, resulting in a descriptor called OSID, invariant to complex brightness change as long as it’s monotonically increasing. Calonder et al. [8] proposed BRIEF, using a binary string consisting the results of pair-wise pixel intensity comparison at pre-determined location, which is very fast in descriptor extraction and matching, but also very susceptible to rotation and scale changes. Rublee et al. [9] improved BRIEF into ORB (Oriented BRIEF), so that it’s rotation invariant and also resistant to noise. There are several other descriptors within this category [11][12][13]. However, this type of descriptor relies heavily on extensive texture, thus they are not suitable for multi-modal matching. The proposed Gixel-based descriptor belongs to the distribution-based category.

3. Introduction to Gixel

A Gixel, or “Gradient Pixel”, is a sample point for the gradient information in a small local region on an image. A traditional Pixel would capture the pictorial information of its neighborhood, and produce an intensity value or a 3D vector (R, G, B). Likewise, a Gixel would extract and summarize the gradient information of its neighborhood. All line segments or pixelized gradients within the neighborhood of a Gixel are sampled (or scored) to produce a 2D vector of both x and y gradient data.

3.1. Additive Edge Scoring

For each line segment or pixelized gradient (a pixel gradient is considered as a line segment of length 1), a Gixel creates a score, which encodes three elements of information: orientation, length, and distance to the Gixel. The longer a line segment is, the higher it is scored. Similarly, the nearer a line segment is to the Gixel, the higher it is scored. The score is then divided into x and y components according to the orientation of the line segment or gradient.

In addition, sensor noise or inaccurate edge detection usually results in small gaps, zigzags, or other artifacts along the line, as shown in Fig.2(a, b). The design of the scoring function makes it additive, so that a sequence of short line segments achieves a similar score as a long connected line, as shown in Fig.2(c).

The function \( f(x) = 1/(1 + x) \): has the properties that \( f(0) = 1, f(\infty) = 0 \), and \( f(x) \) decreases as the input value \( x \) increases.

The scoring function is constructed from \( f(x) \). Given a line segment’s length \( l \), and distance to the Gixel \( d \), the score is defined as:

\[
Score(l, d) = f(d) - f(d + l) = \frac{1}{1 + d} - \frac{1}{1 + d + l}
\] (1)
The circular array is very helpful in Gixel-5.3, and illustrates the scoring function. Since Gixels will also differentiate lines by their orientations. A line should have more importance to the Gixel score. Thus when a Gixel lies along the extension of a line segment, that is always negative and monotonically increasing, making $Score(l, d)$ decreases as $d$ increases. The score is then broken into $x$ and $y$ components, reflecting the orientation of the line segment.

Finally, to achieve the property that short dashed line segments are scored similarly to a long connected line, we use Manhattan distance for the line segment from the intersection point to the nearest point on the line segment (0 if the intersection point is on the line segment). It might seem that using Manhattan distance causes line segments pointing towards the Gixel to be scored higher than those perpendicular ones. This is actually intended, so when a Gixel lies along the extension of a line segment, that line should have more importance to the Gixel score. Thus Gixels will also differentiate lines by their orientations.

For example, as shown in Fig. 2(e), $AB$ is a line segment, $C$ and $C'$ are two points on $AB$ near each other, $G$ is the Gixel, and $F$ is the intersection point from $G$ to $AB$. We want to show that $AB$ and $AC + C'B$ contribute similar scores to the Gixel $G$:

$$Score_{AB} = Score(AB, GF + FA) = f(GF + FA) - f(GF + FB)$$

(2)

$$Score_{AC} + Score_{C'B} = Score(AC, GF + FA) + Score(C'B, GF + FC') = f(GF + FA) - f(GF + FC') + f(GF + FC') - f(GF + FB) + f(GF + FA) - f(GF + FB) = Score_{AB}$$

(3)

The two short line segments $AC$ and $C'B$ will be scored similarly to the whole line segment $AB$, overcoming the impact of the gap $CC'$. This way, we aggregate the impact of multiple end-to-end short segments for improved matching of multi-modal images. Lastly, note that the scoring system is smooth and does not introduce thresholds or discontinuities as a function of segment parameters.

### 3.2. Circular Gixel Array

All line segments or pixelized gradients within the neighborhood of a Gixel will be scored, summed up, and split into a 2-D vector with $x$ and $y$ components to encode orientation information. However, a single 2D vector will not provide enough discriminative power for a complex image. Therefore, for a complete descriptor, we put several Gixels in a region, compute their gradient scores individually, and connect their scores into a larger vector. The vector is then normalized to encode relative gradient strength distribution among the Gixels, as well as reduce the effect of illumination change.

While many Gixel array distributions are possible, we put the Gixels in a circular array to form the complete descriptor, which is named Gixel Array Descriptor (GAD), as shown in Fig. 3(a). The circular array is very helpful in achieving rotation and scale invariance (Sec. 5.3). The Gixels are placed on concentric circles with different radii extending from the center Gixel. There are three parameters involved: the number of circles, the number of Gixels on each circle, and the distance between each circle (Figure 3 shows an example of 2 circles and 8 Gixels on each circle). The circular layout is similar to the DAISY descriptor [14], but the computation of the two descriptors are completely different since we aim at different applications.

Figure 2. (a) Small gap and zigzag artifacts appeared in many edge images. (b) Small gap appeared in edge detection. (c) Additive edge scoring makes short dashed line segments achieve a similar score as a long connected line. (d) Score of a line segment given its length $l$ and distance $d$ to the Gixel. (e) Illustration of Manhattan distance.

Figure 3. (a) Circular Gixel array - each yellow point is a Gixel. (b) A Gixel array can be adjusted for rotation and scale invariance.
The array parameters are determined empirically, but are not critical in our experience. A dense Gixel array may lead to redundancy and unnecessary complexity, as Gixels close to each other will sample similar gradient information. On the other hand, too few or too sparse a Gixel array will reduce the discriminative power of the descriptor, and result in low feature dimension or low correlation between Gixels. In our experiments, we use fixed parameters of 3 circles, 8 Gixels on each circle, and the distance between each circle is 50 pixels (which means the radius is 50, 100, 150 from the center Gixel). Thus there are $3 \times 8 + 1 = 25$ Gixels in total, resulting in a feature dimension of 50.

A circular Gixel array also has several other benefits. It’s easy to generate and reproduce. Each Gixel in the array samples the region evenly. Most importantly, it can be adapted to achieve rotation and scale invariance by rotating Gixels on the circle and adjusting circle radius (refer to Sec. 5.3), as shown in Fig. 3(b).

### 3.3. Descriptor Localization and Matching

Unlike descriptors such as SIFT or SURF, which require a keypoint detection step, GAD can be computed with any detector or at any arbitrary location, while still maintaining good performance even under multi-modal situations, due to its smoothness property (Sec. 4.1). In practice we find that a simple keypoint detection step like the Harris corner detector can be used to identify areas with gradients to speed up the matching process.

The distance between two descriptors is computed as the square root of squared difference of the two vectors. Two descriptors on two images are considered matched if the nearest neighbor distance ratio [1] is larger than a threshold. The nearest neighbor distance ratio is defined as the ratio of distance between the first and the second nearest neighbors.

### 4. Analysis of Advantages

#### 4.1. Smoothness

The design of GAD takes special care to achieve its smoothness property, both in how gradients are scored, and in how the neighborhood size of each Gixel is determined. The scoring function is defined as a smooth function monotonically decreasing with length and distance. The neighborhood of each Gixel is cut off smoothly, so that the score is negligible at its boundary, resulting in a circular neighborhood of 50-pixel radius. A small change in a line segment’s length, orientation, distance or the location of a Gixel sample will only result in a small change in the descriptor. Moreover, the additive scoring property of a Gixel also contributes to the smoothness by alleviating the impact of noise and line gaps.

When matching images of a single modality, it is desirable to use a score function with a sharp peak. Smoothness may not be an important issue in this case. However, for multi-modal matching, even corresponding regions may not have exactly the same intensity or gradient distribution. The smoothness property of a Gixel makes it more robust to noise and thus better suited for multi-modal applications. In addition, a Gixel-based descriptor relies less on accurate selection of a keypoint position. This means that keypoint selection is not critical. Even when keypoints have small offsets between two images, which is common in multi-modal matching, Gixels can still match them correctly.

On the other hand, traditional descriptors doesn’t exhibit smoothness, as they usually divides pixels and gradients into small bins and makes numerous binary decision or selections. Comparison-based descriptors are even worse, because noise or keypoint variations might change the pixel position and ordering completely.

Figure 4 shows the distribution of matching similarity score between two descriptors in corresponding regions (of Fig. 5), when one of the descriptor is moving. The Gixel descriptor (Fig. 4(a)) demonstrates better smoothness and distinctiveness (single peak) than SURF (Fig. 4(b)).

#### 4.2. Multi-Modal Matching

Line features are the most important and sometimes the only available feature in multi-modal matching problems. Each Gixel in the descriptor array samples and scores the line features in its neighborhood from a different distance and orientation. In other words, the Gixels sample overlapping regions of edge information, but from different locations. This spatial distribution of samples imparts the final descriptor with its discriminating power.

On the other hand, traditional distribution-based descriptors tend to break a long line feature into individual grid or bin contributions. Noisy line rendering and detection might impact the distribution statistics heavily, making the matching process unreliable.

Descriptors based on pixel-wise comparison may also suffer in multi-modal matching, as the lack of texture will not provide enough distinctive pixel pairs to work robustly.
5. Experiments

5.1. Evaluation Method

The proposed descriptor is evaluated on both public test data and multi-modal data of our own. For public test data, we choose the well-known dataset provided by Mikolajczyk and Schmid [1], which is a standard dataset for descriptor evaluation used by many researchers. It contains several image series with various transformation, including zoom and rotation (Boat), viewpoint change (Graffiti), illumination change (Leuven), JPEG compression (Ubc). On the other hand, multi-modal matching is still a relatively new research area, so there is no established test dataset yet. We use some of our own data to evaluate the performance of GAD. These data sets will be posted online for others to use in comparisons.

We use SIFT [6], SURF [2], BRIEF [8] and ORB [9] for comparison during our evaluation. They are good representatives of existing descriptors, since they cover both categories described in Sec. 2, and are usually compared to other descriptors, producing similar performance. To make fair comparisons, we also use SURF’s feature detector to determine keypoints locations for GAD. After feature extraction, all descriptors are put through the same matching process to determine matched pairs (Note that GAD, SIFT, SURF all use L2 norm distance, but BRIEF and ORB have to use hamming distance). We use the latest build of Open-SURF\(^1\) for implementation of SURF. For SIFT, BRIEF and ORB, we use the latest version of OpenCV\(^2\). Edges are extracted by a Canny detector.

To evaluate the performance objectively, we computed recall and 1 – precision based on the number of correct matches, false matches, and total correspondence [1]. Two keypoints are correctly matched if the distance between their location under the ground-truth transformation is below a certain threshold. Correspondence is defined as the total number of possible correct matches. By changing the matching threshold, we can obtain different values of recall and 1 – precision. A performance curve is then plotted as recall versus 1 – precision.

5.2. Multi-Modal Matching

Figure 5 shows an important matching problem we encountered during our research. We need to match aerial images to 3D wireframes for urban models texturing. One of the images contains only the building rooftop outlines generated from 3D models, which is a binary image without any intensity or texture information. Figure 5(a,b,c,d) are the matching results of SIFT, SURF, BRIEF, ORB, respectively. None of these establish enough good matches to even apply RANSAC. Figure 5(e) shows the matching result obtained by GAD.

---

\(^1\)http://www.chrisveansdev.com/computer-vision-opensurf.html

\(^2\)http://opencv.willowgarage.com/wiki/
Figure 6. Performance comparison when matching a photo and a drawing. (a) Matching with SIFT. (b) Matching with SURF. (c) Matching with BRIEF. (d) Matching with ORB. (e) Matching with GAD. (f) recall vs. 1 − precision curves.

Figure 7. Performance comparison when matching an intensity image and a depth image. (a) Matching with SIFT. (b) Matching with SURF. (c) Matching with BRIEF. (d) Matching with ORB. (e) Matching with GAD. (f) recall vs. 1 − precision curves.

tained by GAD. Most matches are correct, thus the incorrect ones can be filtered out via RANSAC, as shown in Fig.5(g). Using the refined matches, we can estimate the true transformation homography between the two images, and reproject the aerial images onto the 3D wireframes for texturing, as shown in Fig.5(h). Figure 5(f) provides the recall vs. 1 − precision curve comparison, which shows that traditional descriptors barely works with different texture distribution patterns, as can be seen from their low recall value in the curves, while GAD exhibit robust performance.

Figure 6 demonstrates another possible application of multi-modal matching, to match an ordinary image with an artistic image for image retrieval. One image is a photo of the Statue of Liberty, and the other one is a drawing of the Statue. Figure 6(a,b,c,d) are the matching results of SIFT, SURF, BRIEF, ORB, respectively. None of them exhibits many good matches, while GAD can find a good number of correct matches, in spite of completely different image modality and the slight viewpoint change between the two images, as shown in Fig.6(e). Figure 6(f) provides the recall vs. 1 − precision curve comparison.

Figure 7 shows two images taken on the same area, but
one is an intensity image, and the other is a depth image, with visibly different visual patterns. Figure 7(a,b,c,d) are the matching results of SIFT, SURF, BRIEF, ORB, respectively. None of them manages to find enough correct matches, if any. On the other hand, GAD is able to find many correct matches, as shown in Fig. 7(e). Figure 7(f) provides the recall vs. 1 − precision curve comparison.

The GAD has many other potential multi-modal matching applications, such as matching maps to aerial images, or medical image registration [15], as shown in Fig. 8(a,b).

5.3. Rotation and Scale Invariance

The GAD is easily adjusted to achieve both rotation and scale invariance. The smoothness property of Gixels introduced in Sec.4.1 makes it robust to small scale or rotation changes, as shown in Fig.9(a). This allows a search for scale and rotation changes. The Gixel sample point positions are simply adjusted for each search step in scale or rotation.

As introduced in Sec.3.2, Gixels are organized in a circular array, with several Gixels placed evenly on each circle. To align the descriptor into a new orientation, we just rotate the array w.r.t. the center Gixel, so that all Gixels are still sampling corresponding regions when the center position is matched. In addition, for each Gixel, the score in both x and y components \((x_{old}, y_{old})\) is converted through the triangle function transformation to account for the rotation change into \((x_{new}, y_{new})\):

\[
\begin{align*}
x_{new} &= |x_{old} \times \cos\alpha + y_{old} \times \sin\alpha| \\
y_{new} &= |y_{old} \times \cos\alpha - x_{old} \times \sin\alpha|
\end{align*}
\] (4)

Scale invariance can be achieved similarly. First adjust the radius of each circle in the circular formation according to the scale. Then adjust size of the neighborhood for each Gixel so it covers the same region under different image size. Finally, distance and length value of all gradients within the neighborhood are also scaled linearly. It should be noted that we are not just compensating for a known rotation or scale change. Instead, we search for possible changes, and compensate for each searched change using the methods described above.

Figure 9(b) uses two images from the “Boat” series [1] with both rotation and scale changes, though the 2nd one is further rendered with a “pencil drawing effect” via Photoshop. GAD exhibits rotation and scale invariance by finding a good number of correct matches.

5.4. Public Dataset

Figure 9(a,b) both come from the public dataset [1], where GAD can achieve good performance even under multi-modal situations. We present two more examples in Fig.10(a,b), the “Ubc” series with JPEG compression and the “Leuven” series with illumination change, again both from the public dataset [1]. Due to space limitation, the matching results of other descriptors are not listed here (except their curves). They can be found in previous literatures.

Figure 10(a) shows that the GAD outperforms other descriptors on the “Ubc” data with JPEG compression. A possible explanation is that matching images under JPEG compression to normal images is similar to a multi-modal problem, as JPEG compression will produce more “blob-like” areas with quantization edges and deteriorated texture quality. The smoothness property (Sec.4.1) of Gixels also
likely helps in reducing the impact of JPEG compression. In Fig.10(b) with illumination change, the GAD has a recall rate slightly inferior to other descriptors, but still finds a large number of correct matches with almost no errors.

5.5. Processing Time

GAD’s computation process is time-consuming compared to state-of-the-art descriptors, but no efforts at optimization have been made yet. For examples, Fig.1 (size 512x512) takes GAD 8.9 seconds, while SURF needs 0.7s; Fig.10(b) (size 900x600) takes GAD 19.5s, while SURF needs 1.3s. At this point, speed is not a primary concern in our research, but we’ll pursue optimizations in future work.

6. Conclusion

We introduce a novel descriptor unit called a Gixel, which uses an additive scoring method to extract surrounding edge information. We show that a circular array of Gixels will sample edge information in overlapping regions to make the descriptor more discriminative and it can be invariant to rotation and scale. Experiments demonstrate the superiority of the Gixel array descriptor (GAD) for multimodal matching, while maintaining a performance comparable to state-of-the-art descriptors on traditional single modality matching.

The GAD still has some limitations in its current development status. We have put little effort into optimization, so the run time is slow. In addition, though GAD exhibits rotation and scale invariance, large viewpoint changes may reduce performance, and we have not addressed that issue yet. Finally, as a feature built solely on edges, GAD may not perform well in situations where edges are rare. These issues will be investigated in our future work.

References


