Sensing and Recognizing Surface Textures Using a GelSight Sensor

Rui Li
Massachusetts Institute of Technology
rui@mit.edu

Edward H. Adelson
Massachusetts Institute of Technology
adelson@csail.mit.edu

Abstract

Sensing surface textures by touch is a valuable capability for robots. Until recently it was difficult to build a compliant sensor with high sensitivity and high resolution. The GelSight sensor is compliant and offers sensitivity and resolution exceeding that of the human fingertips. This opens the possibility of measuring and recognizing highly detailed surface textures. The GelSight sensor, when pressed against a surface, delivers a height map. This can be treated as an image, and processed using the tools of visual texture analysis. We have devised a simple yet effective texture recognition system based on local binary patterns, and enhanced it by the use of a multi-scale pyramid and a Hellinger distance metric. We built a database with 40 classes of tactile textures using materials such as fabric, wood, and sandpaper. Our system can correctly categorize materials from this database with high accuracy. This suggests that the GelSight sensor can be useful for material recognition by robots.

1. Introduction

Understanding surface properties is important for humans and robots, and the sense of touch is an important source of information. Tactile sensing can involve many kinds of information, including temperature, slip, vibration, etc.. In this paper we focus on information about local surface geometry, which we will call surface texture here.

To extract surface texture, it is important to have a touch sensor that is compliant (i.e., is soft like a human finger) and has good spatial resolution. Many existing techniques suffer from incompliance [12, 13] or poor spatial resolution [11] compared to human fingers. The recently developed GelSight sensor [1] is built from soft elastomer, and by using computer vision techniques it offers unprecedented levels of spatial resolution.

A GelSight sensor delivers a detailed height map of the surface being touched, in the form of a function $z(x,y)$, where (x,y) are the point coordinates. This can be treated as an image, and the interpretation can be approached as a vision problem. Fig. 1(a) shows a photo of a piece of denim; Fig. 1(b) shows the height map derived by GelSight, displayed as a surface plot. Fig. 1(c) shows the height map displayed as a gray image. Fig. (d) – (f) show a photo of a piece of sandpaper, the height map derived by GelSight displayed as a surface plot and the height map displayed as a gray image.

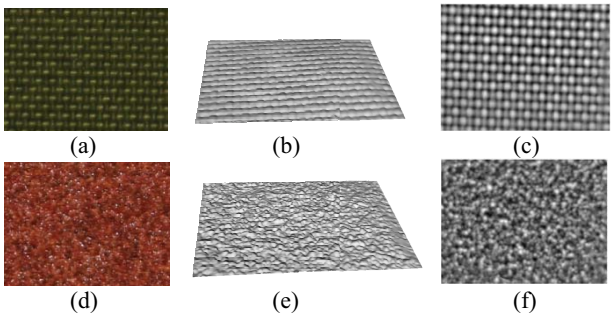


Figure 1: (a) A 2-dimensional (2D) photo of a piece of denim. (b) GelSight height map of the denim rendered at a different view. (c) 2D gray image of the denim height map in (b) with brightness corresponding to the height levels. (d) A 2D photo of a piece of sandpaper. (b) GelSight height map of the sandpaper rendered at a different view. (f) 2D gray image of the height map in (e) with brightness corresponding to height levels.

The problem of tactile texture recognition has some specific properties of note. Many vision problems are simplified, since a height map involves no confounds with shading, albedo, distance, etc.. The spatial scale is fixed, since the sensor is in direct contact with the surface and the camera inside the GelSight device is at a fixed orientation and distance with respect to the sensor.

In most cases, the orientation of the texture will be unknown. For example, a denim texture might occur with an arbitrary rotation, so we want a recognition system that is rotationally invariant. The recognition problem is inherently statistical. For example, every patch of 220-grit sandpaper will look different (at the pixel level) than every other patch of 220-grit sandpaper. Thus we are confronted with a classical texture recognition problem. We cannot recognize the sandpaper with a simple template match; rather, there are certain image statistics that will characterize the sandpaper even though each patch is slightly different.

1.1. GelSight Overview

The GelSight sensor is a novel tactile sensor to capture surface geometry through the use of a gel and a camera that gives a "sight" with computer vision algorithms. It consists of a piece of clear elastomer coated with a reflective membrane. When an object is pressed against the membrane, the membrane deforms to take the shape of the object's surface, which is then recorded by a camera under illumination from LEDs located in different directions. A 3-dimensional (3D) height map of the surface can then be reconstructed with a photometric stereo algorithm [1]. Fig. 2 illustrates the GelSight principle.

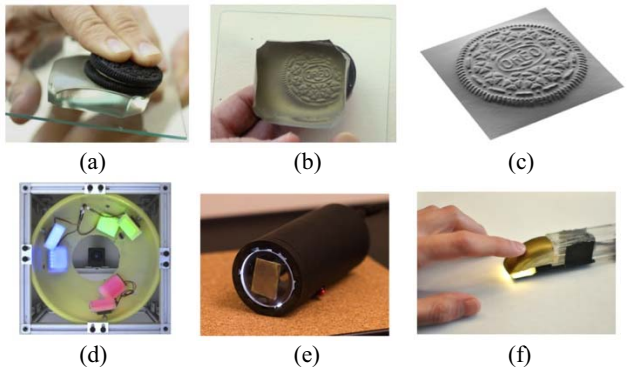


Figure 2 [1]: GelSight illustration. (a) A cookie is pressed against the membrane of an elastomer block. (b) The membrane is deformed to the shape of the cookie surface. (c) The shape of the cookie surface is measured using photometric stereo and rendered at a novel viewpoint. (d), (e), and (f) are the box, portable and finger configurations of the GelSight devices.

The GelSight sensor has many important properties that make it attractive for use in tactile sensing. The sensor is made with inexpensive materials, and it can give spatial resolution as small as 2 microns. In addition, the sensor is not affected by the optical characteristics of the materials being measured as the membrane supplies its own bidirectional reflectance distribution function (BRDF). This allows us to capture a wide range of material surfaces no matter whether they are matte, glossy, or transparent (see Fig. 1). Last but not least, with compliant properties of the gel sensor, GelSight may be used to measure the roughness and texture of a touched surface, the pressure distribution across the contact region, as well as shear and slip between the sensor and object in contact. All these properties make GelSight a very promising candidate to be used in robotic fingertips for tactile sensing.

Height maps obtained using GelSight have some special characteristics. To some extent, GelSight images are sensitive to the amount of force applied. Even for the same surface, with slightly different forces, the gray-scale images can have different gray levels due to different levels of deformations of the gel and/or texture surface. Furthermore, the relative orientation between the gel and the texture can be different for each measurement. Those two characteristics make it desirable to have a texture classification algorithm that is invariant to both gray scales and rotation. We next give an overview of texture classification techniques and discuss what may be used in recognizing tactile textures with the use of GelSight.

1.2. Texture Classification Overview

Texture images are generally spatially homogeneous and consist of repeated elements, often with some random variations (e.g., random positions, orientations, and/or colors). There are generally three types of methods adopted for rotation invariant texture classification: statistical, model-based and structural methods. This work focuses on the statistical methods due to the statistical properties of textures. In statistical methods, the feature distribution method is among the most popular. Pietikainen et al. [10] described texture images using features like center-symmetric auto-correlation, local binary pattern (LBP), and gray-level difference, which are locally invariant to rotation. They propose a feature distribution method based on the G statistics to test those features for rotation-invariant texture analysis. Ojala et al. [2] extended the work by using multiresolution gray-scale and rotation invariant LBP at circular neighborhoods of different radius and neighbor density, and achieved a relatively high classification rate. This had then become the state-of-the-art method, based on which a number of improved texture classification algorithms were developed. Among those are LBP histogram Fourier features (LBP-HF) [6], LBP variance (LBPV) with global matching [7], dominant LBP (DLBP) [8]. Yet one common issue of all these LBP-based methods is that they mostly deal with microstructures of texture images by considering patterns within a small neighborhood (e.g., up to 3 pixels away) but not macrostructures with a large neighborhood.

In this work, we propose a multi-scale local binary pattern (MLBP) operator that can capture both micro- and macrostructures with the use of pyramid levels. Also, we discuss the Hellinger similarity metric for classification. Section 2 describes the traditional LBP, followed by MLBP in Section 3. Section 4 presents the experiment results on Outex databases and GelSight texture images. Section 5 is the conclusion and future work.

2. Local Binary Pattern

LBP [2] is a texture operator for gray-scale and rotation invariant texture classification. It characterizes local structure of the texture image by considering a small circularly symmetric neighbor set of P members on a circle of radius R. The neighborhood is thresholded at the gray value of the center pixel into a binary pattern, which is then weighted by a binomial factor and summed to obtain the

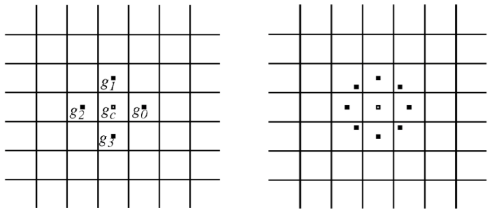
LBP value:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p \quad (1)$$

where

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0, \end{cases} \quad (2)$$

$LBP_{P,R}$ is the LBP value, g_c is the gray value of the center pixel of the local neighborhood, g_p $(p = 0, \dots, P-1)$ correspond to the gray values of the P equally spaced pixels on a circle of radius R $(R > 0)$ [2]. If the coordinates of g_c are (0,0), then the coordinates of g_p are $\left(R \cdot cos\left(\frac{2\pi p}{P}\right), R \cdot sin\left(\frac{2\pi p}{P}\right)\right)$. The gray values of neighbors that do not fall exactly in the center of pixels are estimated by interpolation. Fig. 3 shows the neighborhoods with different P and R values.



(P=4,R=1.0) (P=8,R=1.0)

(a) (b)

Figure 3 [2]: Illustration of local binary patterns with P equally distributed members on a circular neighborhood of radius R. (a) P = 4, R = 1. g_c is the gray value of the center pixel of the local neighborhood, g_p $(p = 0, \dots, P-1)$ correspond to the gray values of the P equally spaced pixels on a circle of radius R $(R > 0)$. (b) P = 8, R = 1.

Signed differences $g_p - g_c$ are not affected by changes in monotonic changes in gray values of pixels; hence LBP is invariant to monotonic gray-scale shifts.

Rotation invariance is achieved by assigning a unique identifier $LBP_{P,R}^{ri}$ to each rotation-invariant local binary pattern, i.e.,

$$LBP_{P,R}^{ri} = \min\{ROR(LBP_{P,R}, i) \mid i = 0,1, \dots, P-1\}, \quad (3)$$

where *ROR(x,i)* performs a circular bit-wise right shift on the P-bit number x i times. The superscript *ri* denotes rotation invariance.

Furthermore, Ojala et al. [2] defined uniformity measure U as the number of spatial transitions (bitwise 0/1 or 1/0 changes) in the pattern, and designated "uniform" patterns as those with U not more than 2. The rotation invariant "uniform" LBP operator $LBP_{P,R}^{riu2}$ is defined as:

$$LBP_{P,R}^{riu2} = \begin{cases} \sum_{p=0}^{P-1} s(g_p - g_c) & if\ U(LBP_{P,R}) \leq 2 \\ P+1, & otherwise, \end{cases} \quad (4)$$

where

$$U(LBP_{P,R}) = |s(g_{P-1} - g_c) - s(g_0 - g_c)| + \sum_{p=1}^{P-1} |s(g_p - g_c) - s(g_{p-1} - g_c)| \quad (5)$$

and the superscript *riu2* reflects the rotation invariance "uniform" pattern with U not more than 2. In practice, $LBP_{P,R}^{riu2}$ has P + 2 distinct output values. The texture feature employed is the histogram of the operator outputs accumulated over a texture sample. $LBP_{P,R}^{riu2}$ is invariant to gray scales and rotation, making it a potentially good candidate for classifying GelSight texture images.

3. Multi-scale Local Binary Pattern

The "uniform" patterns in LBP are indications for structures such as spots, flat areas, and edges of varying positive and negative curvatures. The choice of P and R affects directly the size of the structures under investigation. Intuitively, the larger the R is, the larger the size of the patterns examined; a small R corresponds to microstructures and a large R macrostructures. As discussed in [2], however, P and R are closely related and practically limited by requirements of efficient implementations. Firstly, on one hand, for a given R, we want a large P to reduce the quantization level of the neighborhood circle which is determined by $360°/P$. On the other hand, circular neighborhood for a given R contains a limited number of pixels (e.g., 8 for R = 1), which sets an upper limit for P in order to avoid redundancy in calculating the LBP value. A sensible relationship between P and R is that P = 8R. Secondly, an efficient implementation with a lookup table of 2^P elements sets an upper limit to P for real-time applications [2]. For example, with (P,R) = (32,4), the size of the lookup table for $LBP_{P,R}^{riu2}$ can be up to 4 Gigabytes which is quite big and it becomes slow to find a particular match of the LBP value in the lookup table. This may limit the potential application of the algorithm for real-time implementations. For the above two reasons, Ojala et al. [2] only considered (P,R) values of (8,1), (16,2) and (24,3). However, this limits the capabilities of using macrostructures with R > 3 as texture features with larger P and R. In fact, many texture images in the real world may contain similar microstructures but different macrostructures. Fig. 4 shows an example of two visually very different textures that have similar microstructures but very different macrostructures.

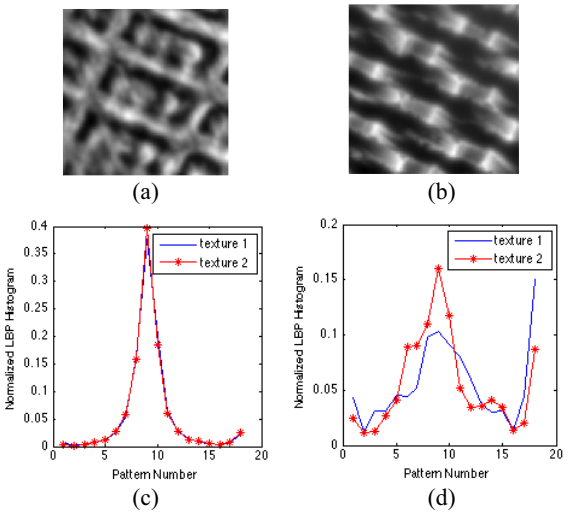


Figure 4: Illustration of two textures with similar microstructures but very different macrostructures. (a) Texture 1. (b) Texture 2. (c) Histogram of LBP values for $(P,R) = (16,2)$ on the original images of texture 1 and 2. This represents statistics of the microstructures with $R = 2$. (d) Histogram of LBP values for $(P,R) = (16,2)$ on images with reduced dimensions at level 2 of Gaussian pyramid (refer to Section 3.1). Equivalently this represents statistics of the macrostructures with $R = 8$ in the original image.

To achieve a high classification rate, it is practically desirable to have operators that include statistics of both microstructures and macrostructures as the texture features without increasing the values of P and R. In the next section, we introduce MLBP that takes into consideration both micro- and macrostructures, and achieve a significant increase in performance.

3.1. The Algorithm

As discussed in Section 2, the conventional LBP may not deal with macrostructures effectively for $R > 3$. Instead of increasing P and R on the original texture image as Ojala et al. did [2], we can first reduce the dimensions of the images and then apply the same P and R to obtain the LBP features. We employ the Gaussian pyramid [3] approach for size reduction before applying LBP.

Denote the original image as pyramid level 0, reduced pyramid level as level 1 and so on. Effectively, reducing image dimensions by half before applying LBP is approximately equivalent to applying LBP with twice the P and R on the original image in terms of normalized LBP histograms (regardless of and the weights used to generate the Gaussian pyramid), yet with a much faster speed. For instance, the histogram information contained in the LBP values obtained with $(P,R) = (16,2)$ at level 1 is approximately equivalent to that obtained with $(P,R) = (32,4)$ at level 0 after normalizing the LBP histograms. By considering histograms of LBP values at different pyramid levels of the original image, we can take into consideration both micro- and macrostructures of different sizes. The feature vector used for classification is a weighted concatenation of LBP histograms at different levels or different image scales. We refer this histogram operator as multi-scale local binary pattern, or MLBP for short.

While the multiresolution LBP by Ojala et al. [2] also tries to combine statistics of structures of different sizes, its fundamental limitation is that it does not go beyond $R > 3$ due to the reasons discussed in the beginning of Section 3 and, therefore, does not take into consideration statistics of macrostructures. The MLBP operator is different from the multiresolution LBP by Ojala et al. [2] in that we find LBP histograms of different pyramid levels with possibly the same P and R for each level, instead of increasing P and R directly on the original image. In this way, it really enables us to consider statistics of structures at different scales. More specifically, we use n different pyramid levels with parameter $a = 0.375$ in Burt and Adelson paper [3] and obtain the histogram of $LBP_{P,R}^{riu2}$ as in Eqn. (4) for each level $i = 0,\ldots, n\text{-}1$, which we denote as $L_{P,R}^{i}$ for simplicity. We then apply weights w_i to $L_{P,R}^{i}$ and concatenate them to form a big feature vector denoted as:

$$MLBP_{P,R}^{n} = concat\left[w_i \cdot L_{P,R}^{i}\right] \qquad (6)$$

for $i = 0,\ldots,n\text{-}1$, where n is the number of pyramid levels, and concat[.] is the concatenation operator. Note that different from previous notations where LBP refers to the pattern over a particular neighborhood of pixels, MLBP here is used to denote the concatenated *histograms* of $LBP_{P,R}^{riu2}$ at different pyramid levels.

For each level reduction, the image size is reduced approximately by half in each dimension, and the number of pixels is reduced to approximately 1/4 of the original image. So is the sum of the LBP histogram. We use the simple weights $w_i = 2^i$ to partially compensate for the reduction of the image size while noting that smaller images play less important, but not negligible, roles in texture classification. For example, for $n = 4$, and $(P,R) = (16,2)$, we have

$$MLBP_{16,2}^{4} = \left[L_{16,2}^{0} \quad 2L_{16,2}^{1} \quad 4L_{16,2}^{2} \quad 8L_{16,2}^{3}\right] \qquad (7)$$

It is possible to use a different set of weights, e.g., based on statistics of the texture database, but we do not see significant improvement over the simple scheme above, while adding in a new class might change the weights completely. Hence we do not go into the detail in this work. Also, an extension to the MLBP operator is to use different (P,R) values at different pyramid levels. We do not see added advantages of doing so, and here we use the same (P,R) value across different levels.

3.2. Similarity Metric

For histogram features, Arandjelovic et al. [4] recently proposed that the Euclidean distance measure often yields inferior performance compared to using measures such as Hellinger. In this work, we used Hellinger distance as the similarity metric. The Hellinger kernel, or Bhattacharyya coefficient [5], for two $L1$ normalized histograms, $\boldsymbol{x}$ and $\boldsymbol{y}$, is defined as:

$$BC(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{n} \sqrt{x_i y_i} \tag{8}$$

where $\sum_{i=1}^{n} x_i = 1$ with $x_i \geq 0$, and $\sum_{i=1}^{n} y_i = 1$ with $y_i \geq 0$.

It was shown in [4] that after the following two steps, comparing Euclidean distances of the resulting vectors is equivalent to comparing Hellinger distances of the original vectors: (i) $L1$ normalize the feature vector so that $\sum_{i=1}^{n} x_i = 1$ with $x_i \geq 0$; (ii) square root each element x_i. This is because the Euclidean distance can then be expressed as:

$$d_E\left(\sqrt{\boldsymbol{x}}, \sqrt{\boldsymbol{y}}\right)^2 = \left\|\sqrt{\boldsymbol{x}} - \sqrt{\boldsymbol{y}}\right\|_2^2 \tag{9}$$

$$= 2 - 2BC(\boldsymbol{x}, \boldsymbol{y})$$

$$= 2H(\boldsymbol{x}, \boldsymbol{y})^2 \tag{10}$$

where $H(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{1 - BC(\boldsymbol{x}, \boldsymbol{y})}$ is the Hellinger distance. In this way, we have the flexibility to apply many readily available built-in functions in various image processing software such as MATLAB. The smaller the Hellinger distance, the more similar the two histograms or feature vectors are.

4. Experiments

Experiments were done on both Outex databases [5] and GelSight images to test the performance of the MLBP operator. The Outex databases contain 2D texture images that are used to compare performance of MLBP with that of other methods. The GelSight images are of real interest for tactile sensing and are used to validate the performance of MLBP. Here we convert GelSight 3D height maps to 2D gray images by using brightness levels to represent the height information. While there is a clear distinction between 2D visual textures such as those in the Outex databases and the 3D surface textures in GelSight, the basic principle of texture classification remains the same.

4.1. Experiment on Outex Databases

The Outex database is a publicly available framework for experimental evaluation of texture analysis algorithms [5]. There are a number of test suites available. We are particularly interested in the following two that are most popular for evaluating texture classification algorithms in terms of invariance to gray scales and rotation:

1. Test suite Outex_TC_00010 (TC10): There are 24 textures in total, and each texture contains 180 samples at nine rotation angles (0°, 5°, 10°, 15°, 30°, 45°, 60°, 75°, and 90°). Each sample has dimension 128×128 pixels. Fig. 5 shows the 24 textures at angle 0°. The classifier is trained with the reference textures (20 samples of illuminant "inca" and angle 0° in each class), while the other 160 samples of the same illuminant but the other eight rotation angles in each texture class, are used for testing the classifier. In this suite, there are 480 training samples and 3,840 testing samples in total.

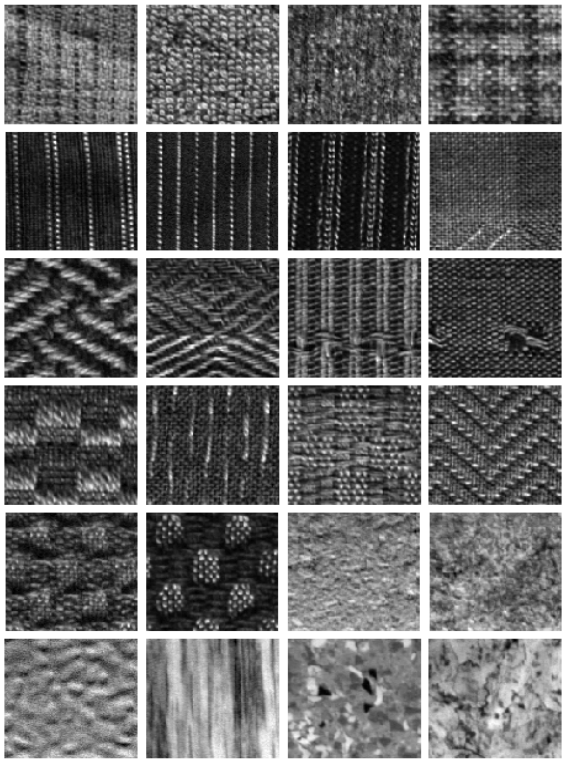


Figure 5: 128×128 samples of the 24 textures in Outex_TC_00010 and Outex_TC_00012.

2. Test suite Outex_TC_00012 (TC12): The classifier is trained with the reference textures (20 samples of illuminant "inca" and angle 0° in each class) and tested with all samples captured using illuminant "tl84" and "horizon". In each problem, there are 480 training samples and 4,320 testing samples.

Our goal is to maximize the classification rate, defined as the ratio of the number of correctly classified samples to the total number of samples for classification. First, we find the MLBP feature vector as in Eqn. (6) for each training sample, $L1$ normalize it and square root each element. For a given testing sample, we do the same operation and find its

3 nearest neighbors among the training samples using Euclidean distance measure. Remember from Eqn. (10) that equivalently we are using Hellinger distance metric on the MLBP. Among the 3 nearest neighbors found, if at least two of them belong to the same class, we output that class as the class of the testing sample. If the 3 nearest neighbors belong to 3 different classes, then we output the class of the nearest neighbor as the class of the testing sample. We use the same pair of $(P,R) = (16,2)$ at $n = 4$ different pyramid levels for this experiment.

Table 1: Correct classification rates (%) of different methods, with the highest rate of each column highlighted in bold. Note that $(P,R) = (16,2)$, and $n = 4$.

	TC10 "inca"	TC12 "tl84"	TC12 "horizon"
$\mathrm{LBP}^{\mathrm{riu2}}_{\mathrm{P,R}}$	96.10	88.80	83.40
$\mathrm{LBP}^{\mathrm{riu2}}_{\mathrm{P,R}}/\mathrm{VAR}_{\mathrm{P,R}}$	97.70	87.30	86.40
LBP-HF	97.97	91.50	87.66
$\mathrm{LBPV}^{\mathrm{u2}}_{\mathrm{P,R}}\mathrm{GM}_{\mathrm{ES}}$	97.76	95.39	95.57
DLBP	99.10	93.20	90.40
MR8	92.50	90.90	91.10
MLBP	**99.17**	**98.91**	**98.22**

Table 1 shows the correct classification rate using different methods by comparing MLBP with 6 other classical texture classification algorithms: $\mathrm{LBP}^{\mathrm{riu2}}_{\mathrm{P,R}}$ [2], $\mathrm{LBP}^{\mathrm{riu2}}_{\mathrm{P,R}}/\mathrm{VAR}_{\mathrm{P,R}}$ [2], LBP-HF [6], LBPV with global matching ($\mathrm{LBPV}^{\mathrm{u2}}_{\mathrm{P,R}}\mathrm{GM}_{\mathrm{ES}}$) [7], dominant LBP (DLBP) [8], and MR8 [9]. LBP-HF, $\mathrm{LBPV}^{\mathrm{u2}}_{\mathrm{P,R}}\mathrm{GM}_{\mathrm{ES}}$, and DLBP are improved versions of LBP, and their best performances among all (P,R) pairs used by the authors are listed here for comparison. MR8 is the state-of-the-art statistical algorithm for texture classification.

From Table 1 it can be seen that for all TC10 and TC12 databases under different illuminations, MLBP achieves the best classification rates among all 7 methods compared and is most invariant to different illuminations. In particular, for the TC10 database, MLBP increases the rate to 99.17% from 96.10% of $\mathrm{LBP}^{\mathrm{riu2}}_{\mathrm{P,R}}$. For the TC12 "tl84" and "horizon" databases, MLBP increases the rate by 10.11% and 14.82% respectively, compared to $\mathrm{LBP}^{\mathrm{riu2}}_{\mathrm{P,R}}$. This shows that the MLBP operator is most invariant to rotations under the same illuminant. Furthermore, when we compare the performance of MLBP for TC10 and TC12 under 3 different illuminants, we see that the classification rate is very stable (98.22% ~ 99.17%) as compared to other methods such as DLBP. This means that MLBP is also invariant to gray scales to a large extent. Nevertheless, when the illumination is not uniform, which is often the case in real-world conditions, all the above methods may not perform well simply due to the fact that shadows or illuminants now become part of the textures. Hence it becomes beneficial for us to use GelSight height images combined with MLBP to classify those textures.

4.2. Experiment on GelSight Images

We obtained 40 classes of GelSight texture images from the GelSight portable device (Fig. 2(e)) using the techniques described in [1]. Each class consists of 6 texture images at random orientations and with dimensions 480×640 pixels. Each image is then cropped to 4 non-overlapping samples of dimension 200×200 pixels at the center of the image, with 960 samples in total.

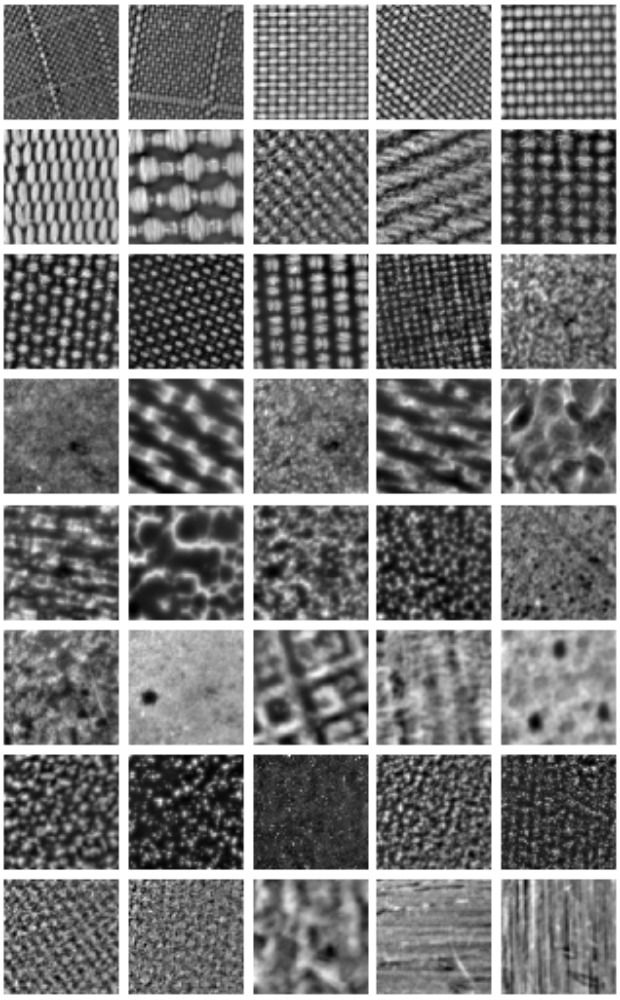


Figure 6: GelSight texture database with 40 different texture classes, comprised of, from left to right and up to down, 14 fabrics, 13 foams, 3 tissue papers, 7 sandpapers, 1 plastic and 2 wood textures.

The actual images obtained from GelSight are height maps. We convert them to 2D images with brightness of pixels indicating the height levels: the brighter the pixel in the corresponding 2D image, the larger the height is. Fig. 6

shows samples of the 40 texture classes. Numbered from left to right and up to down, the surfaces are 14 fabrics, 13 foams, 3 napkins, 7 sandpapers, 1 plastic and 2 woods. Note that the database contains some really similar textures, such as textures 1 and 2, 17 and 19, 15, 16, 18, and 25, etc., which makes the classification task challenging. Among the 24 samples for each texture class, some are used as the training samples and the rest as testing samples. Table 2 shows the correct classification rate for different numbers of training and testing samples. Here we use $n = 4$ pyramid levels.

Table 2: Correct classification rate (%) of MLBP for different numbers of training samples with the highest rate highlighted in bold.

Number of training samples per texture	MLBP $(P,R) = (16,2)$	MLBP $(P,R) = (8,1)$
8	99.22	98.12
12	**99.79**	98.96
16	99.69	99.38

It can be seen that among all the different settings, MLBP with $(P,R) = (16,2)$ and 12 training samples can give the best performance of 99.79% with only one sample misclassified out of 480 testing samples. As the number of training samples increases, the correct classification rate is expected to increase as well, as there are more samples to be compared with. But the classification speed may become to decrease. In practice, we will find a compromise between the number of training samples used and speed especially when the classification is performed in real time, such as in the case of robotic tactile sensing. Also, we may use different (P,R) pairs for different tasks.

5. Conclusion

Tactile sensing is an important but challenging area for robotics. With the compliant properties of gel elastomers that mimic human fingers, GelSight is a promising candidate for tactile sensing and material perception. This work focuses on the classification of surface textures, where the texture data is based on height maps attained by touching a surface with a GelSight sensor. We adopted techniques based on local binary patterns (LBP). Conventional LBP and improved versions such as LBP-HF and DLBP mainly look at microstructures of textures and overlook the macrostructures that may be important distinguishing features for different textures. In this work, we presented a novel multi-scale operator, MLBP, that takes into consideration both microstructures and macrostructures for feature extraction. We also adopted the Hellinger distance as a similarity metric. To compare our algorithm with current techniques in the visual texture literature, we used the Outex databases. MLBP performed the best among several classical methods for texture classification. We also built a database of GelSight surface textures, with 40 classes of different materials, and achieved a classification rate as high as 99.79%. Although the database is small, the high classification rate indicates that our system is well suited to the task of recognizing high-resolution surface textures, and may help to deliver a rich form of information for robotics.

References

[1] M.K. Johnson, F. Coley, A. Rajz and E. H. Adelson, "Microgeometry Capture using an Elastomeric Sensor", SIGGRAPH 2011.

[2] T. Ojala, M. Pietikainen and T. Maenpaa, "Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, pp.971−987, 2002.

[3] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," IEEE Trans. Commun., vol. 31, no. 4, pp. 532-540, April 1983.

[4] R. Arandjelović, A. Zisserman. "Three things everyone should know to improve object retrieval," IEEE Conference on Computer Vision and Pattern Recognition, 2012

[5] T. Ojala, T. Maenpaa, M. Pietikainen, J. Viertola, J. Kyllonen, and S. Huovinen, "Outex – A New Framework for Empirical Evaluation of Texture Analysis Algorithms," *Proc. 16th Int'l Conf. Pattern Recognition*, 2002.

[6] T. Ahonen, J Matas, C. He and M. Pietikainen. Rotation invariant image description with local binary pattern histogram Fourier features. *SCIA Proc.*, pages 61-70, 2009.

[7] Z. Guo, L. Zhang, and D. Zhang. Rotation invariant texture classification using LBP variance (LBPV) with global matching. Pattern Recognition, pages 706-719, 2010.

[8] S. Liao, M. W. K. Law, and A. C. S. Chung, "Dominant Local Binary Patterns for Texture Classification," IEEE Transactions on Image Processing, vol. 18, no. 5, May 2009.

[9] M. Varma and A. Zisserman. A statistical approach to texture classification from single images. *International Journal of Computer Vision: Special Issue on Texture Analysis and Synthesis*, 62(1--2): 61--81, April 2005.

[10] M. Pietikainen, T. Ojala, Z. Xu, Rotation-Invariant texture classification using feature distributions, Pattern Recognition 33 (2000) 43–52.

[11] Fishel, J.A., Loeb, G.E., Bayesian exploration for intelligent identification of textures. *Frontiers in Neurorobotics* 6:4, 2012

[12] R. M. Crowder. Toward robots that can sense texture by touch. Science, 312:1478–1479, 2006.

[13] V. Maheshwari and R. F. Saraf. High-resolution thin-film device to sense texture by touch. Science, 312:1501–1504, 2006.