FEATURE RECOGNITION AND FINGERPRINT SENSING FOR GUIDING A WOOD PATCHING ROBOT

Tobias Pahlberg¹, Olle Hagman²

ABSTRACT: This paper includes a summary of a few commonly used object recognition techniques, as well as a sensitivity analysis of two feature point recognition methods. The robustness was analyzed by automatically trying to recognize 886 images of pine floorboards after applying different levels of distortions. Recognition was also tested on a subset of 5% of the boards which were both re-scanned using a line scan camera and photographed using a digital camera. Experiments showed that both the Block matching method and the SURF method are valid options for recognizing wood products covered with distinct features. The Block matching method outperformed the SURF method for small geometric distortions and moderate radiometric distortions. The SURF method, in its turn, performed better compared to the other method when faced with low resolution digital images.

KEYWORDS: Wood, Feature recognition, Fingerprint, Hol-i-Wood PR, Patching robot, Image analysis, Classification, Holonic, Sensor fusion

1 INTRODUCTION

Suppose a wooden board has gotten a crack, a few dead knots have loosened, and maybe the hue of a part of the board has changed. The question is if it is possible for a machine vision system to identify the board using only its biometric “fingerprint”? We may want to trace back through the history of the board. How was it produced and under what conditions? Or perhaps the task is merely to detect if the right piece has arrived at the right machine?

The objective of the presented work is to determine if it might be possible find a certain wooden board in a database of a thousand boards; and to determine a feasible type of descriptor that a computer can use to recognize boards. The descriptors need to be robust to both radiometric and geometrical distortions to be able to recognize a particular wood piece using different cameras and different camera setups.

The Hol-i-Wood Patching Robot project started in early 2012 and is a collaboration between Luleå University of Technology (LTU), TU Wien and TU München, as well as industrial partners; MiCROTEC, Springer, TTTech and LIP BLED. The project will have resulted after three years in several different standalone holonic modules of which the wood fingerprint recognition is one part.

The hardware will consist of a new multi-sensor scanner, a system for transport of laminated wood products which need repairing, and lastly a patching robot. The patching robot will be equipped with a vision system and should be able to recognize previously scanned wood pieces at the repair station, preferably using a non-invasive method. The patching robot will drill out certain defects and replace them with fitted wooden dowels. Defects such as dead knots, pitch pockets and geometrical defects. The identification using the fingerprint of wood is needed to inform the patching robot of which piece has arrived.

A number of laminated wooden end-products nowadays depend on a skilled machinist to repair the defects offline, which can become a bottleneck in the production chain.

2 THEORY

The purpose of this chapter is to summarize and describe a few important concepts related to identification systems.

2.1 FINGERPRINTS

Within the field of biometrics the human fingerprint has been used for identifying people for over a century [1]. Scanning technology and algorithms have been pushed forward mainly by security and forensics applications. The procedures in fingerprint recognition software most
often have to address the following four design challenges [2]:

1. Image acquisition
2. Fingerprint representation
3. Feature extraction
4. Matching

Human fingerprint systems often rely on extracting certain minutiae or details for identifying individuals [1]. Ridge endings, ridge bifurcations and singular points are usual descriptors together with the coordinates and the orientation of these descriptors. Similar techniques can likewise be applied to identify wooden boards, where for example their knots and growth ring patterns describe “individuals” in the same way as the ridges do on a finger.

2.2 ROBUSTNESS

It can be very difficult to acquire an image under exactly the same conditions at different scanning stations. Furthermore, a lot of things can have happened to an object over time since a previous scan. Hence, a practical identification system needs to be robust to noise and distortions. Geometric distortions can arise if for example a wooden board is transported skew past a line scan camera, if the pixels are non-square, or if the cameras have different board is transported skew past a line scan camera, if the pixels are non-square, or if the cameras have different orientations. Moreover, radiometric distortions introduced by different lighting conditions could be present, and at some manufacturers, for example, sawdust or dirt can occlude parts of the boards.

2.3 SPEED

A common way to reduce the problem of quickly having to search through a huge database of fingerprints to find a match is to group similar prints into bins [1]. These bins can be used for classification purposes if desired, but they can also be used to reduce the search space in a matching stage. Bin sorting methods can on the other hand lead to problems if noise is present, where some features can become ambiguous. In such a case the sought board might have been put into the wrong bin and is therefore impossible to find. Instead of sorting boards into discrete classes, a feature vector can be used to represent it. The Euclidean distance can then be calculated between the query vector and vectors in the database.

2.4 CLASSIFIERS

Neural networks (NN) are widely used as a classification method for fingerprints. A fingerprint system called PCASYS which utilizes NNs was developed for the FBI in the early 1990s [3]. Feature extraction and classification of wood defects was done using NNs in [4, 5]. The learning problem in those cases was solved by the use of self-organizing maps (SOM) in the early feature extraction stages. A correct classification rate of 86%, for example, was obtained using Gabor filters and adaptive color histograms. A tree-structure support vector machine (SVM) approach was proposed in [6] to classify four types of wood knots, achieving an average correct classification rate of 96.5%. In a related thesis work a supervised classifier was trained with Adaboost to classify different stain type defects on wood in [7]. Different classifiers have different strengths and weaknesses; hence a combination of classifiers is often the best solution.

2.5 OBJECT RECOGNITION

Object detection, classification and recognition are tightly intertwined areas. The sought output is different for them, but the process of getting the output is usually similar. Often the procedure involves extraction of feature points from the objects. A feature point is a region in an image that is likely to be recognized in other images [8]. Typical feature points include, for example, corners, blobs and T-junctions. Corners, i.e., positions in an image where there is a strong intensity change in two orthogonal directions, are very good objects to track [9, 10]. A region around the corner is usually stored. These regions, or templates, are later used for recognition of objects by finding several matching regions between images. Criteria for good feature points, or interest points, are described in [11]:

1. Distinctness: The points should be distinguishable from their neighborhood, e.g., consist of a pronounced gradient in intensity or color.
2. Invariance: The points should be invariant with respect to expected geometric and radiometric distortions.
3. Stability: The points should be robust to noise.
4. Seldomness: There should not be several similar points in the same image to avoid confusion. (If a point is part of a repetitive pattern, the possibility for a false match is high.)
5. Interpretability: The points should preferably be interpretable, such as an edge, corner or blob.

Edges are usually not good interest points; the region information is similar along the line and hence does not fulfill the seldomness requirement. Smooth untextured regions do not uphold the distinctness requirement, etcetera. Several corner- and feature point-detection algorithms have been produced over the years. One of the most famous corner detectors is the Harris corner detector [12]. The Harris method involves moving a search window over a grayscale image and computing the weighted Sum of Squared Difference (SSD) between the starting location and its surroundings (Figure 1). In a smooth untextured part of the image there will not be a big SSD response in any direction. At a line, or a sharp gradient change, there will be a big response in...
By use of Taylor expansion, Equation (1) can approximately be rewritten as:

\[ D(x, y) \approx (x, y) A \begin{pmatrix} x \\ y \end{pmatrix}, \]

where

\[ A = \sum_{u,v} w(u,v) \begin{bmatrix} \frac{\partial^2 I(u,v)}{\partial x^2} & \frac{\partial^2 I(u,v)}{\partial x \partial y} \\ \frac{\partial^2 I(u,v)}{\partial y \partial x} & \frac{\partial^2 I(u,v)}{\partial y^2} \end{bmatrix}. \]  

Second derivatives are better suited for finding lines and corners than first derivatives, i.e., gradients. Second derivatives are only non-zero at places in the image where there is a gradient change, not where there is a constant gradient. A constant gradient is usually not defined as a line.

Most corner detectors utilize the Hessian matrix, in some way, to find intensity maxima.

Through the eigenvalues, \( \lambda_1 \) and \( \lambda_2 \), of \( A_{2 \times 2} \), a rotationally invariant descriptor of what is going on around a point in an image is obtained.

1. If both \( \lambda_1 \) and \( \lambda_2 \) are large, a corner is present in the region.
2. If one eigenvalue is large and the other is small, then a line is present in the direction perpendicular to the eigenvector with large eigenvalue.
3. If both \( \lambda_1 \) and \( \lambda_2 \) are small, there is no a distinct edge or corner in that region.
The minimum error, or maximum response, is given when the template is at the correct location.

2.6 GEOMETRIC TRANSFORMATIONS

The geometric relation between points in space and a two-dimensional image can be described by a projective transformation [18]. If most of the points in a scene are at the same depth, and if a long focal length is used, then the affine transformation is a good approximation of the projective transformation. Affinely distorted images can be generated using the planar affine transformation. This 2D to 2D transformation skews the image and then rotates and translates it (Figure 3). The transformation preserves straight lines, i.e., lines which were parallel, will be parallel also after the transformation. Furthermore, affine transformations preserve the length ratios between parallel line segments in an image, as well as ratios between areas.

The planar affine transformation $x' = Hx$ is defined as:

$$
\begin{bmatrix}
  x' \\
  y' \\
  1
\end{bmatrix} =
\begin{bmatrix}
  a_{11} & a_{12} & t_x \\
  a_{21} & a_{22} & t_y \\
  0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  x \\
  y \\
  1
\end{bmatrix} =
A
\begin{bmatrix}
  x \\
  y \\
  1
\end{bmatrix}.
$$

(4)

The linear transformation, $A$ in Equation 4, consists of a deformation followed by a rotation:

$$
A = R(\theta)R(-\phi)D(\phi).
$$

(5)

After the first rotation, $\phi$, the image is scaled in $x$- and $y$- directions using the diagonal matrix in Equation 6, skewing the image.

$$
D = \begin{bmatrix}
  \lambda_1 & 0 \\
  0 & \lambda_2
\end{bmatrix}
$$

(6)

The image is then rotated back by the same amount and a different rotation, $\theta$, is applied. Lastly, a translation by $(t_x, t_y)$ is applied to the image.

3 MATERIALS AND METHODS

3.1 IMAGES AND IMAGE ACQUISITION

The chosen dataset consists of 886 floorboards from Scots pine (Pinus sylvestris) with the dimension 21x137 mm. The same set of floorboards was previously used in [19-21]. They are between 3007-5109 mm long, with a mean length of approximately 4500 mm. The boards were sawn from 222 logs with top diameter between 201-215 mm and were randomly collected from Bollsta sawmill in central Sweden. The boards had been planed, sanded and finished with white pigmented oil and a thin layer of varnish.

The finished floorboards were scanned in 2006 using a high resolution color line scan camera; Dalsa Trillium TR-37. It is a 3CCD camera with a 2048 pixel array for each of the colors; red, green and blue [19]. The boards were scanned at a resolution of 2.5 pixels/mm lengthwise and 10 pixels/mm across. Photocells were set up to automatically crop off 80 mm from each side to avoid edge artifacts. The images were then resized to a resolution of 1 pixel/mm in both directions using bicubic interpolation and were saved in JPEG format at 95% quality. The colors of the images had been carefully calibrated to match the colors of the boards in reality.

3.2 ROBUSTNESS TESTING

To test the robustness of the two different feature-matching techniques, a two-level full factorial experiment was created. A subset of 100 floorboards was picked out randomly from the total 886. The planar affine transformation was used to distort these images using different parameter combinations. The four parameters; $\theta$, $\phi$, $\lambda_1$ and $\lambda_2$, were each assigned a “low” and a “high” value. Hence, $2^4 = 16$ new image datasets containing 100 images each were created to test all combinations and to determine which factors had the most significant impact on the identification accuracy. The different low and high values can be seen in Table 1.

<table>
<thead>
<tr>
<th>$\theta$ (°)</th>
<th>$\phi$ (°)</th>
<th>$\lambda_1$</th>
<th>$\lambda_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>5</td>
<td>5</td>
<td>0.9</td>
</tr>
<tr>
<td>High</td>
<td>15</td>
<td>15</td>
<td>0.7</td>
</tr>
</tbody>
</table>

(7)

The translation parameters of the affine transformation were set to zero since they would not affect the result in any case. Feature points are matched independently of their positions and only their relative positions in each image are used to sort out false matches.

The low and high values were chosen accordingly after screening experiments indicated difficulties within these ranges. Note that when the scaling parameters, $\lambda_1 = \lambda_2 = 1$, there is no scaling done. As can be seen in Table 1, a 10% size reduction is chosen as the low value, and a 30% size reduction as the high value.

Anti-aliasing is used to limit the impact of aliasing on the output images when resizing and transforming images. Aliasing may appear as “stair-step” patterns at lines in an image or as moiré patterns, a ripple- or wave-type artifact.

3.3 RE-SCAN OF IMAGES

The same line scan camera as before was used in 2012 to re-scan 5% of the floorboards to have a subset of realistically acquired images to test the matching capabilities on. Some of the boards had during the past
years of storage become somewhat crooked or bowed. Moreover, no stabilization of the boards was carried out while the conveyer belt was transporting them past the camera at the re-scan. No color corrections were made on the images and some dust stains were present on a few of the floorboards. This time the camera was mounted at such a distance from the conveyer belt that yielded a resolution of 5.6 pixels/mm lengthwise and 6.5 pixels/mm across. The entire lengths of the board images were kept this time, and the images were resized to 1 pixel/mm in both directions using bicubic interpolation. They were saved in JPEG format at maximum quality.

### 3.4 DIGITAL PHOTOS
To provide a last tough challenge to the recognition methods, digital photos were taken of the floorboards from a distance. The photos were acquired with a Canon 400D camera at a resolution 0.6 pixels/mm. The images were scaled up to 1 pixel/mm in order not to disadvantage the scale-dependent Block matching method.

### 3.5 FEATURE EXTRACTION AND MATCHING
Two different methods were used for feature extraction and matching in the experiments. The first is referred to as the Block matching method and the other as the SURF method. Implementations of these methods can be found as open source [22], and are often used for similar purposes in other computer vision applications. In this paper, implementations from MATLAB® R2011b, Computer Vision System Toolbox™, have been incorporated.

#### 3.5.1 Block matching method
In the Block matching method simple square neighborhoods are extracted as features around interesting points in an image. A detailed explanation is presented in Algorithm 1.

A few settings for the FAST corner detector were altered from the default. The size of the search window for finding corners was set to 25x25 pixels, i.e., 25x25 mm due to the resolution. The corner threshold parameter was increased to 0.001 (default: 0.0005) to pass along also weaker corners. However, the intensity threshold for accepting corners was instead increased to 20 (default: 0.1) out of the maximum 127.5, to reduce the number of corners being falsely detected on the boards’ boundaries. No more than the 50 best corners were kept to be able to execute the matching step within reasonable time.

The intensity information in a 25x25 pixel neighborhood around each FAST point was extracted as feature points. This neighborhood size was chosen since it encircled most of the knots. The MATLAB-function `matchFeatures` carries out steps

---

**Algorithm 1: Block matching method**

1. Run FAST corner detector on query image, $I_q$, and database image, $I_b$.
2. **Description of feature points:**
   2.1. Extract the intensity information in a square patch around the corners.
   2.2. Patches that are on the border or completely outside the region of interest are removed.
   2.3. Reshape the patches into $n_x$ and $n_y$ column vectors and insert them as the columns of feature matrices $A$ and $B$.
   2.4. Normalize the columns of $A$ and $B$ to make the method more robust to radiometric differences.
3. **Matching:**
   3.1. Calculate the $(n_x \times n_y)$ error matrix between all the features in $A$ and $B$ using SSD according to:
   $$E(i, j) = \sum ((a_i - b_j)^2),$$
   where $a_i$ and $b_j$ are the column vectors of $A$ and $B$ respectively, and the operator $(\cdot)^2$ squares every element in a vector.
   3.2. For every row in $E$, find the lowest column error, i.e., the best matching features between $I_q$ and $I_b$.
   3.3. Remove duplicate feature pairs.
   3.4. Remove weak matches which have an error above some specified threshold.

---

**Algorithm 2: SURF method**

1. Create a scale-space representation of the query image and the database image by use of integral images and box filters.
2. **Detection of feature points:**
   2.1. Use the local maxima of the Hessian determinant operator applied to the scale-space to obtain feature points.
   2.2. Keep points with a response level above a certain threshold.
   2.3. Refine scale and location of these candidates.
   2.4. SURF points which are on the border or completely outside the region of interest are removed.
3. **Description of feature points:**
   3.1. Calculate a dominant orientation direction using the neighborhood gradient information to make the feature points rotationally invariant.
   3.2. Build a 64-dimensional descriptor corresponding to the local normalized gradient histograms using Haar wavelets.
4. **Matching:**
   4.1. Compute the Euclidean distance between all potential matching pairs.
   4.2. Reduce mismatches using a nearest neighbor distance ratio criterion.
3.1-3.4 in the Block matching algorithm. There, the SSD boundary for accepting matches was increased to 2% instead of the default, 1%, to produce more potential matches. Feature pairs with a matching error above this percentage are removed.

3.5.2 SURF method
The SURF method is thoroughly described in [16], and an extensive algorithm description can be found in [23]. A shortened version is presented in Algorithm 2. The number of octaves was kept at the default, 3, but the number of scale levels within each octave was increased to 6 (default: 4). No maximum number of SURF points was specified here, only the default metric threshold for accepting points was used. The SSD boundary for accepting matches was increased to 5% instead of the default, 1%. The MATLAB-function matchFeatures carries out steps 4.1-4.2 in the SURF algorithm.

3.6 REMOVAL OF FALSE MATCHES
Some features of wood, such as knots, can look quite similar, especially to a computer. Hence, features from one floorboard are sometimes matched to several features in another. This ambiguity can be taken care of by removing features that get a lot of matches (Seldonnness requirement, Section 2.5). Features with more than three possible candidate matches were removed. A problem can also be that the best fit of features, algebraically, might not be the correct one, for different reasons. Hence, it is usually a bad idea to sort out “erroneous” matches solely on the basis of a SSD threshold.

3.6.1 Geometric Transform Estimator
MATLAB’s Geometric Transform Estimator was utilized to remove point pairs that should be considered as false matches given a certain geometric transformation. The non-reflective similarity transformation was chosen instead of the affine transformation since it, due to fewer degrees of freedom, performed more stably. A shortened version is presented in Algorithm 3. To allow for some deviation and to loosen the demands on the point pairs to conform to the model, the Euclidean pixel distance threshold was raised to 100 mm. Random Sample Consensus, RANSAC [24], was chosen as the method for determining inliers, i.e., true matches. A description of the geometric transform estimator is presented in Algorithm 3.

4 RESULTS AND DISCUSSION
In this section the two different feature recognition methods are compared. Note, however, that their performances depend on parameters which do not always apply to both methods. Test runs were carried out on a PC running Windows 7 64-bit with an Intel® Xeon® processor at 2.53 GHz. It takes about 12 seconds to acquire feature points for one query floorboard and then compare it against the database of 886 floorboards, i.e., around 14 ms to compare it against one database-file. The Block matching method and the SURF method take roughly the same amount of time.

4.1 ROBUSTNESS TESTING
For every distorted image, the best matching board in the database was chosen based on having the greatest number of unique matching feature points. Both the Block matching method and the SURF method successfully identified the correct individual in Figure 4. The identification accuracy of the two methods for different parameter combinations can be seen in Table 2. The Block matching method performs well as long as the scaling parameters are kept at the low level. The scaling in x-direction, $\lambda_1$, has the biggest impact on the accuracy. Inspection of some of the failed cases showed that occasionally the FAST corner detector misinterpreted sections of the boards’ outer borders as corners and thus a lot of feature points were rejected. The worst case occurred at a large skewing angle, $\phi$, in conjunction with a large scaling in x-direction, $\lambda_2$. However, neither a large rotation angle nor a large skewing angle seems to correlate significantly with bad accuracy. The SURF method’s identification accuracy remains decent when the scaling parameters are kept at the lower level. However, the SURF method has big trouble when either $\lambda_1$ or $\lambda_2$ is at the high level. A noted problem is that certain wood features that the SURF method sees as especially interesting are found at many different scale levels. Thus, several scattered features in one image can be falsely matched together with a tight cluster of features in the other image. This effect is most evident when either $\lambda_1$ or $\lambda_2$ is at the high level and the other one is not. The SURF method, however, does not experience the same problems of finding invalid feature points on edges of the boards.

A large number of matched feature points usually means that the correct board has been identified. If the second best matching board has almost the same amount of matched feature points, then the chances are high that the “best” might not be the correct one. To visualize the difference between the number of matched feature points for the best match and the second best match, for small geometric distortions, a box plot was created (Figure 5).

---

**Algorithm 3: Geometric transform estimator**

1. Loop until maximum number of iterations has been reached, or until enough inliers have been found.
   1.1. Randomly pick a number of matched point pairs from images $I_a$ and $I_b$ (2 pairs for non-reflective similarity transformation).
   1.2. Calculate the transformation matrix, $H_s$, using the relationship $x' = H_s x$.
   1.3. Project the points from one image onto the other and sum the Euclidean distances between the point pairs.
   1.4. Update $H_S$ if the summed distance is lower than the previous minimum.
2. Return a list of inlier point pairs.
Each of the 886 floorboards was given a small geometric distortion and was then matched against the original images. The parameters of the affine transformation were assigned uniformly distributed random numbers on the intervals: $-5^\circ \leq \theta, \varphi \leq +5^\circ$ and $0.9 \leq \lambda_1, \lambda_2 \leq 1.1$. The plots show that the Block matching method has a slightly higher mean number of matched features for the best match.

### 4.2 RE-SCANNED IMAGES

An example of how one of the re-scanned images could look is shown in Figure 6. Both the Block matching method and the SURF method successfully identified the correct board also in that case. The identification accuracy of the 44 re-scanned boards can be seen in Table 3. The Block matching method outperforms the SURF method for this particular case comprising small geometric distortions and moderate radiometric changes. Due to the smaller geometric distortions in this experiment, the drawbacks of the two methods should not negatively affect the accuracy to the same extent.

**Table 2:** Comparison of the identification accuracy between the Block matching method and the SURF method for the simulated distorted images dataset at different maximum rotation angles and scaling parameters. The best accuracy is marked green for each method and the worst, red.

<table>
<thead>
<tr>
<th>Block matching method</th>
<th>SURF method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy (%)</td>
<td>Accuracy (%)</td>
</tr>
<tr>
<td>93</td>
<td>98</td>
</tr>
<tr>
<td>94</td>
<td>48</td>
</tr>
<tr>
<td>90</td>
<td>36</td>
</tr>
<tr>
<td>78</td>
<td>75</td>
</tr>
<tr>
<td>94</td>
<td>98</td>
</tr>
<tr>
<td>100</td>
<td>47</td>
</tr>
<tr>
<td>68</td>
<td>35</td>
</tr>
<tr>
<td>78</td>
<td>76</td>
</tr>
<tr>
<td>97</td>
<td>92</td>
</tr>
<tr>
<td>98</td>
<td>50</td>
</tr>
<tr>
<td>78</td>
<td>36</td>
</tr>
<tr>
<td>82</td>
<td>78</td>
</tr>
<tr>
<td>100</td>
<td>93</td>
</tr>
<tr>
<td>98</td>
<td>53</td>
</tr>
<tr>
<td>96</td>
<td>28</td>
</tr>
<tr>
<td>82</td>
<td>79</td>
</tr>
</tbody>
</table>
A visualization of the difference between the numbers of matched features for the best match and the second best match is shown for each of the 44 boards in Figure 7. The same tendency as before can be seen there. The SURF method has a more widespread number of matched features, though there is more often a smaller difference between the best and the second best match. This means that the SURF method can be seen as less robust in that sense.

**Table 3:** Comparison of the identification accuracy between the feature point methods for the re-scanned images.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block matching</td>
<td>95.5</td>
</tr>
<tr>
<td>SURF</td>
<td>86.4</td>
</tr>
</tbody>
</table>

### 4.3 DIGITAL PHOTOS

The last and most challenging case was that of matching low resolution digital images with the original, higher resolution, images. The digital images were cropped and scaled to match the resolution of the original images. The identification accuracy is shown in Table 4, where the SURF method is clearly the most robust method in this case.

Closer investigation showed that the FAST corner detector had trouble finding features in these images. Different settings were tested but did not produce more feature points. Hence the somewhat undeserved poor result for the Block matching method.

**Table 4:** Comparison of the identification accuracy between the feature point methods for the digital photos.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block matching</td>
<td>9.1</td>
</tr>
<tr>
<td>SURF</td>
<td>34.1</td>
</tr>
</tbody>
</table>

![Figure 5: The number of matching feature points for the best matching floorboards compared to the second best match when $-5^\circ \leq \theta, \phi \leq +5^\circ$ and $0.9 \leq \lambda_1, \lambda_2 \leq 1.1$. (a) Block matching method. (b) SURF method.](image1.png)

![Figure 6: Example of a re-scanned floorboard image identified in the database. (a) Block matching method. (b) SURF method.](image2.png)
CONCLUSIONS

There are many parameters that can be varied and tweaked to get different amounts of feature points from the algorithms. If thresholds for accepting corners are lowered, more feature points will be returned. However, these will be less robust.

A drawback of having a lot of features is that the extraction and matching procedure will take longer, making a real-time application less feasible.

A problem with the FAST corner detector is that it can in some cases misinterpret jagged lines and edges as corners. The SURF method’s corner detector is in this respect more reliable and better at finding features in an image.

The method of finding interest points using corners, i.e., a high second derivative response in two directions, is a very general method. It is widely used within machine vision in various different scenarios. A wooden surface with knots is a special case that probably should be dealt with using a more specialized knot-finding algorithm. The Block and SURF features are likewise common general purpose feature points, but not very specialized towards recognizing wood. The advantage of these methods is that they are quite robust. It would be easy to re-tune the parameters and apply the methods on a slightly different recognition problem.

The SURF method’s problem of incorrectly matching several scattered features in one image to a tight cluster of features in the other image must be addressed. This is not a huge problem and can definitely be solved.

The Block matching method performed surprisingly well in all experiments, especially considering that it is scale and rotation dependent. The method would, however, greatly benefit from using a more robust knot-finding algorithm.

The rotationally invariant SURF method generally handled rotations worse than the Block matching method, which was unexpected. Rotational invariance is possibly not as crucial due to knot features being mostly rounded.

Both the Block matching method and the SURF method are valid options for recognizing wood products covered with distinct features. The Block matching method outperformed the SURF method for small geometric distortions and moderate radiometric distortions. The SURF method, in its turn, performed better compared to the other method when faced with low resolution digital images. However, the FAST corner detector often failed to acquire enough valid interest points to the Block matching method in this case.

A combination of Block- and SURF features would most certainly provide a big improvement in identification accuracy.

Future work will include the incorporation of multi-sensor data and defect classification used in existing systems for wood inspection.

ACKNOWLEDGEMENTS

The authors would especially like to thank Olof Broman for giving us access to the floorboards used in this work. Also, thanks to Simon Dahlquist at SP Trä for assisting us with the re-scanning of the boards.

REFERENCES

Pattern Recognition and Artificial Intelligence, 10:97-113, 1996.


