MVA2011 IAPR Conference on Machine Vision Applications, June 13-15, 2011, Nara, JAPAN

A Technique for Clustering Individual Defects from Images of Steel Strips with Periodical Defects

Francisco G. Bulnes, Daniel F. Garcia, Ruben Usamentiaga and Julio Molleda
Department of Computer Science, University of Oviedo
Campus de Viesques, 33204, Gijon, Spain
bulnes@uniovi.es

Abstract

The steel strips produced in steel-making plants, are used as raw material in many other industries, so quality control is an essential aspect. One factor that indicates the quality of a steel strip is the number of defects, such as holes or scratches, on its surface. This paper describes a technique to detect an especially harmful type of surface defect called periodical defect. These defects are a periodic pattern on the surface of the strip. Using a backtracking algorithm all the individual defects contained in the strip are examined to determine which defects compose a set which constitutes a periodical defect.
An implementation of this clustering technique was tested using a set of real steel strips, whose characteristics were previously stored in a database. A test environment to quantify the goodness of the results and to determine the best values to parameterize the clustering algorithm has also been developed.

1 Introduction

During the rolling of steel strips, periodically repeated defects can occur. Such defects are particularly detrimental to the quality of a steel strip, because they do not consist of a single mark on the surface of the strip, but of many marks along the length of the strip, creating a periodic pattern.

These defects are characterized as being a series of marks in the same transversal coordinate, being equally spaced at a constant distance in the longitudinal coordinate, and having the same shape. A set of these defects is called a periodical defect. It is important to detect the existence of these defects as soon as possible to prevent their occurrence in the next steel strips to be rolled. If these defects are not detected, the steel strips may be unsuitable for marketing. Having effective software to perform this task will not only improve steel strip quality, but will also avoid the financial loss caused by having to discard defective steel strips.

2 Industrial Context

To understand how periodical defects are generated and how they can be detected, it is necessary to know some aspects of the steel rolling process in steel mills.

The strips are derived from large pieces of steel, called slabs, extracted from a furnace at a high temperature (about $1200°C$). Then, the steel passes through a rougher mill, where pressurized water ($180Kg/m^2$) is applied to the strip to remove the scale generated during heating. Next, the steel travels along a roll path until it reaches the finishing mill, where seven stands are serially arranged to apply pressure to the strips one after another. The steel is stretched and flattened into a strip of great length. Then, while dragging the steel strip on another roll path, it is cooled by applying water curtains. Finally, the steel strip is rolled to form a coil.

Each stand of the finishing mill has a pair of work rolls, which press the hot steel. The state of these rolls is critical because they are in direct contact with the steel, applying high pressure. Rolls with a damaged surface produce periodical defects.

It is imperative to replace the rolls that are imprinting defects as soon as possible. A damaged work roll marks the steel strips, producing periodical defects with a constant period. The final goal of the technique presented in this work is to determine which roll of which stand is causing the periodical defects, using the longitudinal and transversal coordinates of all the individual defects on the surface of a strip. This task must be done quickly (within one minute) in order to provide the result before starting the rolling of the next strip.

3 Detection of Individual Defects

To determine which of the individual defects are caused by a defective work roll, the position of all the defects on both the top and the bottom surfaces of each strip must be identified. This task is performed by an external computer vision system, which inspects the entire surface of both sides of the steel strips.

Images captured by each camera are processed in real time, in order to detect individual defects (see Figure 1). The position (longitudinal and transversal) of each defect and the area it occupies, are stored in a database for further processing.

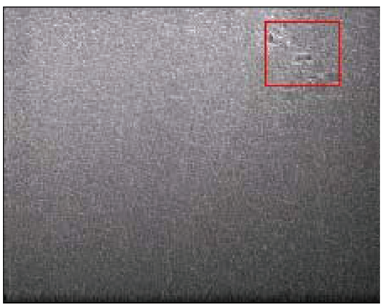

Figure 1. Detection of an individual defect

Once this task is done, the information about each individual defect detected on both surfaces of the strip is available in the database. Now, the task is to determine which defects have been generated by damaged roll, and to quickly cluster them into periodical defects.

4 Clustering of Individual Defects

Once information about the position and size of all the individual defects on the strip is available, it remains only to cluster those that are part of the same periodical defect. To design an algorithm to cluster individual defects into a periodical defect, the features of periodical defects must be defined.

4.1 Defect description

A periodical defect can be defined as a set of individual defects located in the same transversal coordinate of the strip, spaced evenly along the longitudinal coordinate. Surfaces may have more than 2,000 individual defects to examine. To determine whether a number of these defects constitutes a periodical defect, the conditions mentioned in the Introduction must be fulfilled: they are all situated in the same transversal coordinate and equally spaced by a constant distance in the longitudinal coordinate. The search space created is too large to be explored in a short period of time. For this reason, it is crucial to bound it. [1] proposes a method to estimate the period length of periodical defects produced by each of the work rolls, taking into account their radii, work roll spacing and the separation between the rolling stands. The relationship between the thickness reduction of the strip and the increment of its length can be easily established, as shown in Figure 2.

In this example, a damaged work roll generates a periodical defect whose period length is L when the thickness of the strip is S. If the gap between the rolls of the next stand is $S/2$, the thickness of the steel strip passing through it is also $S/2$. By halving its thickness, so that the steel volume remains constant, its length shall be doubled ($2L$). Thus, considering the initial thickness of the steel and the gap between the work rolls of each stand, the period length of the periodical defects produced by each stand can be estimated. With this information, we can avoid the task of finding individual defects repeated with any period. We seek only periodical defects whose period length coincides with the estimated lengths calculated in this manner. This greatly reduces the search space.

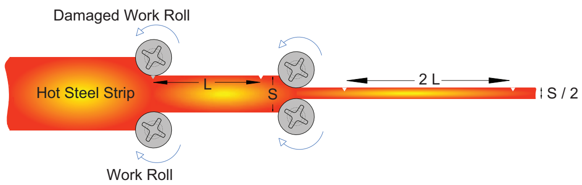

Figure 2. Relation between the reduction of thickness and the elongation of the strip

In most cases only the defects imprinted by the last three work rolls can be detected. The severity of the defects produced by the first rolls is mitigated when the steel is smoothed by subsequent rolls. Thus, for the first four work rolls, only very serious defects can be detected. For this reason, almost all periodical defects have a period length between 2 and $3m$.

4.2 Clustering Algorithm

Before individual defects can be efficiently clustered, they must be stored in memory using an appropriate data structure. In this case, we propose the use of sparse matrices. Thus, the steel strip is treated as a matrix where the longitudinal and transversal coordinates of a single defect may be used as indexes in this matrix. By using the methods of storage of sparse matrices [2], no memory is wasted storing large numbers of elements (zones of strip surface) with no defects.

The algorithm must examine the non-zero elements of the matrix (individual defects), and check if other non-zero elements meet the requirements of a periodical defect. To seek a periodical defect with period length N, the algorithm must check if there are non-zero elements in the positions of the strip whose information is stored in *matrix[i, j]*, *matrix [i+N, j]*, *matrix[i+2N, j]* and so on. This process is repeated until no individual defects are detected in the desired position in the matrix. Once an individual defect is identified as belonging to a periodical defect, it must be labelled as CLUSTERED to avoid including it in a subsequent search. Naturally, for this algorithm to work properly, it must be given some flexibility. Occasionally an individual defect which belongs to a periodical defect cannot be found in a particular position although it is known to exist. This can occur if, when processing images for individual defects, they are placed in slightly different positions.

Figure 3 shows individual defects detected in the previous phase by the computer vision system, corresponding to the same periodical defect. Although the individual defects are identical in shape, the system has generated a rectangle with a different area and position for each of them. This makes it necessary to establish a transversal tolerance and a longitudinal tolerance around the position where the search is to be carried out. Thus a rectangle (also called a search area) is defined to seek the individual defect within it.

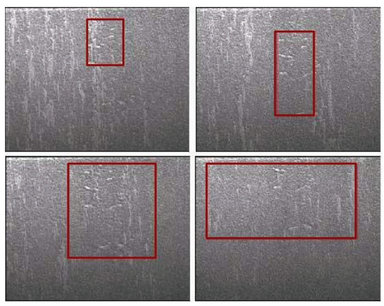

Figure 3. Detection of individual defects

It must also be noted that in strips with a high density of individual defects, two defects that are not generated by the same work roll may be located in positions *matrix[i, j]* and *matrix[i+N, j]* (where N is the theoretical period length of one work roll). If the defect density is high enough, this could happen not only with two individual defects, but with three or four. To prevent these individual defects from being clustered as periodical defects, the algorithm must be parameterized to avoid clustering fewer than a certain number of individual defects. This number should be large enough to make it virtually impossible to cluster non-periodic defects as described above.

Another problem that arises quite often is that when processing images, the computer vision system fails to detect an individual defect. This usually occurs with ill-defined defects or superficial scratches. If the search is stopped when an individual defect is not found, parts of this periodical defect may not be classified. To avoid this, the search algorithm must continue even if a number of individual defects are not detected consecutively.

Thus, 5 parameters are established that will allow some flexibility for the search:

- Min_Defs: Minimum number of individual defects which must to be clustered in a periodical defect.
- Max_Skips: Maximum consecutive individual defects undetected before stopping the search.
- T_Tol: Width of the search area.
- L_Ratio: Ratio between the length of the search area and the theoretical period length of the work roll whose periodical defects are being sought.
- A_Ratio: Ratio between the average area of the individual defects clustered in a periodical defect and the area of an individual defect to be included in the same periodical defect.

As the search space has been largely reduced, a backtracking algorithm [3] can be used to explore it, taking into account the parameters described above. An implementation of backtracking algorithms can be analyzed in [4]. In an attempt to make the classical backtracking algorithm as efficient as possible, some improvements have been introduced to reduce computation time [5]. Combining this with the reduction of the search space [6], the individual defects of a complex strip can be clustered in several seconds. A simplified pseudo code of the algorithm is shown below.

```
for each defect d do
    d.clustered=FALSE
end for
for each workroll D do
    PeriodicalDefect pd
    for each defect d //matrix[x,y] do
        if d.clustered == FALSE then
            Section sec ← d
            Search(x + D, y, sec, Max_Skips)
        end if
    end for
    if sec.size ≥ Min_Defs then
        pd ← sec
    end if
end for

Procedure SearchArea (i, j)
for a = i − T_Tol to i + T_Tol do
    for b = j − L_Ratio to j + L_Ratio do
        if matrix[a, b] ≠ 0 then
            return (a, b)
        end if
    end for
end for
return 0
End Procedure
```

```
Procedure Search (i, j, sec, skips)
if SearchArea(i, j) ≠ 0 then
    if matrix[a, b].clustered == FALSE and
    matrix[a, b].area ≤ A_Ratio ∗ sec.AvgArea then
        sec ← matrix[a, b]
        matrix[a, b].clustered = TRUE
        Search(a + D, b, sec, skips)
    end if
else
    if skips ≠ 0 then
        Search(i + D, j, sec, skips − 1)
    else
        return
    end if
end if
End Procedure
```

5 Evaluation

To evaluate the degree of goodness provided by the algorithm, a set of test strips was examined to seek their periodical defects automatically using the algorithm. The results were compared with those provided by an experienced operator. The results provided by the expert are considered the optimal solution, and are referred to as the "ground truth". The expert clustered the individual defects of each of the test strips manually (using a tool).

Once the ground truth is available, the ideal configuration of the algorithm can be determined. The objective of this task is to determine the most appropriate value for each configuration parameter. The clustering algorithm must be flexible enough to work acceptably with strips of different characteristics.

5.1 Metrics

To qualify each solution (clustering) generated automatically by the algorithm it is necessary to apply some kind of metric. In the survey carried out by [9], metrics are classified into two types: analytical and empirical. Analytical metrics evaluate the algorithms themselves. Empirical metrics evaluate an algorithm by measuring the goodness of its results. This can be done by measuring certain characteristics in the results or by simply comparing them with the ideal result. These latter metrics, called *empirical discrepancy metrics*, are those used to measure the results provided by the proposed technique. These metrics provide values between 0 and 1. A value of 1 is given to a perfect fit and a value of 0 to a completely unacceptable solution.

5.2 Experimental design

For optimum performance of the algorithm, it was necessary to determine the values of the configuration parameters that provide the best results. A classic experimental design [7] could have been developed but due to the limited number of test strips available, a specific procedure was carried out. The experimental phase was divided into two tasks. The objective of Task 1 was to obtain the best configuration for each strip. As a single execution of the clustering algorithm takes less than 5 seconds, an extensive search of the search space of 5 dimensions formed by the set of configuration parameters was feasible. Thus, an optimal configuration for each of the test strips was obtained. The objective of Task 2 was to determine how the metrics vary when each of the parameters are modified.

Thus we were able to identify the most important parameters in the metric, and therefore determine which must be fixed more precisely. To accomplish this task, a new set of executions was performed. Each execution started with the optimal configuration determined in Task 1 for each strip, varying each of the parameters separately. Both tasks were performed using grid technologies, due to the large computational cost they require. Specifically, Condor [10] was used. The experiments were performed by a grid of 30 heterogeneous computers, which handles a total of 60 cores.

5.3 Analysis of results

After completing Task 1, the configuration that provides the best results for each strip was available, as well as the values of their metrics. As the algorithm parameters are continuous values, they were discretized for experimentation. Using Task 2, it was determined that the most sensitive parameters are Min_defs and Max_skips. In order to achieve better results, these parameters were discretized with smaller steps.

Table 1 shows, for each strip, the best result obtained with this technique compared with the results obtained by Parsytec [8], the most widely used tool for defect inspection in steel-making plants.

Table 1. Obtained Results

Strip	Backtracking	Parsytec
Strip 1	0.93	0.82
Strip 2	0.60	0.47
Strip 3	0.86	0.78
Strip 4	0.89	0.75
Strip 5	0.82	0.80
Strip 6	0.87	0.50
Strip 7	0.91	0.76
Strip 8	0.94	0.61
Strip 9	0.87	0.77
Strip 10	0.86	0.72
Strip 11	0.85	0.72
Strip 12	0.81	0.49
Strip 13	0.79	0.38
Strip 14	0.63	0.36
Strip 15	0.51	0.13
Strip 16	0.59	0.29
Strip 17	0.74	0.61
Strip 18	0.71	0.55

The best values for each parameter are different for each strip. Using a geometric mean of the best values obtained for each strip, a set of parameter values that provides good results for all strips was determined. This has been done using a geometric mean of the best values obtained for each strip. Thus, the parameter values that could provide near-zero metric values for some strips are avoided. For each parameter, the values obtained using this method are shown in Table 2.

Using this set of parameter values, the metric values obtained for each strips are lower than those shown in Table 1. However, the results are still better than those obtained by Parsytec.

Parameters Max_Skips and A_Ratio must have very large values due to the low accuracy of the data provided by the artificial vision system.

Table 2. Optimal parameters

Parameter	Value
Min_Def	8
Max_Skips	32
T_Tol	42mm
L_Ratio	5%
A_Ratio	500%

6 Conclusions

The work presented in this paper proposes a technique to detect periodical defects in hot steel strips. Sparse matrices are used to store information efficiently, and a vision-based backtracking algorithm is proposed to defect periodical patterns in hot steel strips. They can be detected using this algorithm in a few seconds using a maximum of 20MB of memory.

This algorithm was optimally configured in order to maximize the quality of its detection of periodical defects. A set of test strips was used to test the proposed technique. For all the strips tested the results achieved are better than those obtained by a commercial tool used worldwide.

References

[1] S. Zhao, H. Huang, F. Wang, Y. Li, G. Wei. "Estimating periodic length and defect coordinates form a series of integer values," *Measurement*, vol. 42, pp.516-523, 2009.

[2] I.S. Duff, A.M. Erisman and J.K. Reid. "Direct Methods for Sparse Matrices," *Oxford Science Publications*, 1986.

[3] J.R. Bitner and E.M. Reignold. "Backtrack Programming Techniques," *Communications of the ACM*, vol. 18(11), pp. 651-655, 1975.

[4] K. Helsgaun. "CBack: A Simple Tool for Backtrack Programming in C," *Software Practice and Experience*, vol. 25(8), pp. 905-934, 1995.

[5] P. W. Purdom. "Search rearrangement backtracking and polynomial average time," *Artificial Intelligence*, vol. 21(1-2), pp. 117-133, 1983.

[6] W.L.G. Koontz, P.M. Narendra and K. Fukunaga. "A Branch and Bound Clustering Algorithm," *IEEE Transactions on Computers*, vol. 24(9), pp. 908-914, 1975

[7] D.C. Montgomery. "Design and Analysis of Experiments," John Wiley & Sons, New York, 1997.

[8] J. H. Jeong and M. Kalowitsch. "State-of-the-art surface inspection technology with Parsytec 5i," *SEAISI (South East Asia Iron and Steel Institute)*, vol. 36(1), pp. 10-15, 2005.

[9] Zhang, Y. "A survey on evaluation methods for image segmentation," *Pattern Recognition*, vol. 29(8), pp. 1335-1346, 1996.

[10] D. Thain, T. Tannembaum and M. Livny. "Distrubuted computing in practice: The condor experience," *Concurrency Computation Practice and Experience*, vol. 17(2-4), pp. 323-356, 2005.