Time Series Classification
using Compressed Recurrence Plots

Thilo Michael, Stephan Spiegel, Sahin Albayrak
DAI-Lab, Berlin Institute of Technology,
Ernst-Reuter-Platz 7, 10587 Berlin, Berlin
{thmichael, spiegel, albayrak}@dai-lab.de

Abstract. In recent years recurrence plots have become a widely accepted tool for identifying and visualizing structural patterns in time series. It has been shown that the structural patterns found in recurrence plots can be used to determine the similarity between two time series, which is necessary for classification. For instance, it has been proposed to employ video compression algorithms for measuring the similarity between two recurrence plots, which visualize the structural patterns that were extracted from the time series under study.

In this work we assess to what extent the choice of video compression algorithm influences the similarity measurements and classification performance for recurrence plots or time series respectively. Furthermore, we introduce a novel time series distance measure based on the compression of cross recurrence plots. Our evaluation shows that more advanced compression algorithm do not necessarily result in higher classification accuracy, but lead to superior results for relatively long time series.

1 Introduction

Classifying time series has become an important task in many computer science disciplines like data mining and knowledge discovery [4, 12]. An important part of this classification is the definition of a similarity function that measures the distance of two time series under study. In [2] Campana and Keogh proposed a distance measure that employs the video compression algorithm MPEG-1 to determine the similarity of textures. In [10] Silva et al. applied the video compression distance to recurrence plots and showed that especially the classification of shape-based time series benefit from the new approach.

While the MPEG-1 standard proved to be a good basis for the compression of recurrence plots, it is unclear how different and more advanced video encoding algorithms contained in standards such as MPEG-2 and MPEG-4 effect the distance measure and thus the classification.

In this work we further analyze the possibilities of applying the concept of video compression to recurrence plots. We analyze how well the recurrence plot compression distance performs with newer video compression algorithms such as...
MPEG-2 and MPEG-4. We also propose a new compression distances that combines the concepts of video compression distances and cross recurrence plots. In our experimental evaluation we analyze how the different approaches perform with the different video compression algorithms.

2 Background

2.1 Recurrence plots

Recurrence plots (RP) are used to visualize and analyze systems with nonlinear behavior. The input data is usually represented as a vector of real values, which correspond to different states of the system under study. The recurrence plot illustrates recurring states as individual points and recurring segments as line structures [8].

Fig. 1: The plot of a time series (1) showing CO$_2$ in the atmosphere measured every month between 1958 and 1975 [3]. Below that are the resulting unthresholded recurrence plot (2) and thresholded recurrence plot with a threshold of 0.2 (3).

A recurrence plots of a time series $x$ can be mathematically described as follows:

$$R_{i,j}^{d,t} = \Theta(\epsilon - ||x_i - x_j||) \quad x_i \in \mathbb{R}^d, i, j = 1 \ldots n$$
where $\epsilon$ is a threshold distance, $|| \cdot ||$ is a norm, and $\Theta(\cdot)$ is the Heaviside step function. According to this definition, a recurrence of a state at time $i$ at a different time $j$ is pictured within a two-dimensional squared matrix with black and white dots, where black dots mark a recurrence [7].

Unthresholded recurrence plots are a variation where the threshold parameter $\epsilon$ and the Heaviside function are removed, which results in gray-shaded plots. Figure 1 compares traditional recurrence plots with unthresholded recurrence plots with the help of a sample time series.

Extending the concept of recurrence plots, cross recurrence plots (CRPs) are used to visualize recurring patterns between two time series $x$ and $y$:

$$CR_{i,j}^{d,\epsilon} = \Theta(\epsilon - ||x_i - y_j||) \quad i, j = 1 \ldots n, 1 \ldots m$$

A cross recurrence plot (CRP) shows all those times at which a state in one dynamical systems occurs in a second dynamical system. In other words, the CRP reveals all the times when the trajectories of the first and second time series, $x$ and $y$, visits roughly the same area in the phase space. The data length, $n$ and $m$, of both systems can differ, leading to a non-square CRP matrix [9, 11].

### 2.2 Campana-Keogh distance

The Campana-Keogh distance CK-1 is a Kolmogorov complexity-based compression distance for textures. It uses the MPEG-1 algorithms to calculate a semimetric distance [2].

The MPEG-1 encoding is a specific set of algorithms used to compress a series of coherent images. It not only uses similarities in the horizontal and vertical data of the images (intra frame compression) but heavily relies on similarities in temporal space (inter frame compression) [6].

When encoding a series of images with MPEG-1, each images is either stored as an intra-frame (often called I-frame) or as a predictive frame (often called P-frame). The compressed I-frames can be seen as keyframes that contain all the information to reconstruct the images while the P-frames are storing the information that describes the transition from the previous image to the next one. So to decompress a P-frame, the decoding algorithm has to decompress the preceding I-frame and subsequent P-frames up to the current frame [6].

CK-1 makes use of the MPEG-1 I-frames and P-frames to measure the distance between two images. As a prerequisite the two images are converted to grayscale to provide color invariance and are scaled to the same size. Then an MPEG-1 video is created with two frames. The first frame is an intra frame of the first image and the second frame is a predictive frame of the second image. Relying on the inter frame compression of MPEG-1, Campana and Keogh reason that
the size of the compressed video is getting smaller as the images get more similar [2].

The CK-1 distances measure is defined as follows:

\[ CK1(x, y) = \frac{C(x|y) + C(y|x)}{C(x|x) + C(y|y)} - 1 \]

where \( x \) and \( y \) are the time series to compare and \( C(x|y) \) is a function that evaluates the size of a two frame long MPEG-1 file with \( y \) as the I-Frame and \( x \) as the P-Frame [2].

2.3 Compression distance on recurrence plots

In recent work [10] Silva et al. employed the CK-1 distance to measure the similarity between unthresholded recurrence plots that were generated from time series. They argue that the MPEG video compression algorithm is able to detect similar structures in recurrence plots, which correspond to similar time series patterns. Their experimental evaluation of this recurrence plots compression distance (RPCD) shows that, in comparison to the euclidean distance and dynamic time warping, the combination of the CK-1 distance measure together with unthresholded recurrence plots results in higher classification accuracy for time series which represent shapes (e.g. leaves or faces that are encoded into time series).

3 Proposed method

In the following we introduce our novel cross recurrence plots compression distance (CRPCD), which combines the concept of the discussed compression distance with cross recurrence plots (CRPs). Our proposed approach computes the compression distance between the RPs of two individual time series and their corresponding CRPs.

The intuition behind our proposal is that CRPs are able to reveal co-occurring patterns, which are assumed to be advantageous for video encoding algorithms. Our hypothesis is that the introduced video compression approach produces better classification results when using CRPs, since these plots reveal patterns that co-occur within two time series.

To simplify the notation of the CRPCD, we will denote a recurrence plot \( R^x \) with the equivalent cross recurrence plot \( CR^{x,x} \).

For the calculation of the cross recurrence plots compression distance of two time series \( x \) and \( y \), the plots \( CR^{x,x} \), \( CR^{y,y} \), \( CR^{x,y} \) and \( CR^{y,x} \) have to be generated. With a video compression algorithm the distance between \( CR^{x,x} \) and \( CR^{x,y} \) and
between $CR^{x,y}$ and $CR^{y,x}$ can be calculated as defined by the CK-1 algorithm and then combined to form one distance measure.

Mathematically, the CRPCD measure can be defined as follows:

\[
CK^2(\mathbf{x}, \mathbf{y}) = \frac{C(CR^{x,x} | CR^{x,y}) + C(CR^{y,y} | CR^{y,x}) + C(CR^{y,x} | CR^{x,y})}{C(CR^{x,y} | CR^{x,y}) + C(CR^{y,x} | CR^{y,x}) + C(CR^{x,y} | CR^{x,y}) + C(CR^{y,x} | CR^{y,x})} - 1
\]

where $x$ and $y$ are two time series, $CR^{a,b}$ is the cross recurrence plots of the time series $a$ and $b$ and $C(A|B)$ is a compression function for object $A$ given object $B$.

4 Experiments

In our experimental evaluation we investigate: i) how the accuracy of RPCD [10] changes when we use other compression algorithms (such as MPEG-2 and MPEG-4); and ii) how our proposed CRPCD measure performs in comparison.
4.1 Experimental setup

We used publicly available libraries [1] to test the RPCD method with the newer compression algorithms MPEG-2 and MPEG-4. We configured the libraries to use modes that favor inter-frame to intra-frame compression, but we did not have the resources to perform an exhaustive search over all possible parameter settings. Hence, there is still potential for further improvement.

<table>
<thead>
<tr>
<th>Data set</th>
<th>ED</th>
<th>DTW</th>
<th>RPCD MPEG1</th>
<th>RPCD MPEG2</th>
<th>RPCD MPEG4</th>
<th>RPCD CRPCD MPEG1</th>
<th>RPCD CRPCD MPEG2</th>
</tr>
</thead>
<tbody>
<tr>
<td>5Words</td>
<td>63.10</td>
<td>69.00</td>
<td>73.36</td>
<td>73.85</td>
<td>61.76</td>
<td>78.46</td>
<td>71.43</td>
</tr>
<tr>
<td>Adiac</td>
<td>61.10</td>
<td>60.40</td>
<td>67.47</td>
<td>70.08</td>
<td>61.38</td>
<td>79.88</td>
<td></td>
</tr>
<tr>
<td>Beef</td>
<td>53.30</td>
<td>50.00</td>
<td><strong>63.33</strong></td>
<td><strong>63.33</strong></td>
<td><strong>63.33</strong></td>
<td>46.67</td>
<td>46.67</td>
</tr>
<tr>
<td>ChlorineConcentration</td>
<td><strong>65.00</strong></td>
<td>64.80</td>
<td>51.09</td>
<td>63.78</td>
<td>??</td>
<td>49.93</td>
<td>63.33</td>
</tr>
<tr>
<td>CucC ECG torso</td>
<td>89.70</td>
<td>65.10</td>
<td><strong>97.90</strong></td>
<td><strong>97.24</strong></td>
<td>94.20</td>
<td>93.19</td>
<td>86.16</td>
</tr>
<tr>
<td>Coffee</td>
<td>75.00</td>
<td>82.10</td>
<td><strong>100.00</strong></td>
<td><strong>100.00</strong></td>
<td><strong>100.00</strong></td>
<td>85.71</td>
<td>85.71</td>
</tr>
<tr>
<td>Cricket X</td>
<td>57.40</td>
<td><strong>77.70</strong></td>
<td>70.77</td>
<td>38.46</td>
<td>13.59</td>
<td>75.64</td>
<td>53.59</td>
</tr>
<tr>
<td>Cricket Y</td>
<td>64.40</td>
<td>79.20</td>
<td>73.85</td>
<td>42.31</td>
<td>11.03</td>
<td><strong>82.56</strong></td>
<td>52.05</td>
</tr>
<tr>
<td>Cricket Z</td>
<td>62.00</td>
<td><strong>79.20</strong></td>
<td>70.77</td>
<td>40.76</td>
<td>12.82</td>
<td>77.69</td>
<td>56.41</td>
</tr>
<tr>
<td>DiatomSizeReduction</td>
<td>93.50</td>
<td><strong>96.70</strong></td>
<td>96.41</td>
<td>94.77</td>
<td>95.42</td>
<td>96.08</td>
<td>96.08</td>
</tr>
<tr>
<td>ECG200</td>
<td><strong>88.00</strong></td>
<td>77.00</td>
<td>87.00</td>
<td>88.00</td>
<td>81.00</td>
<td>88.00</td>
<td>86.00</td>
</tr>
<tr>
<td>ECGFiveDays</td>
<td>79.70</td>
<td>76.80</td>
<td>86.41</td>
<td>87.92</td>
<td>88.27</td>
<td>80.48</td>
<td><strong>90.36</strong></td>
</tr>
<tr>
<td>FaceAll</td>
<td>71.40</td>
<td>80.80</td>
<td><strong>80.95</strong></td>
<td>73.96</td>
<td>11.24</td>
<td>80.59</td>
<td>79.74</td>
</tr>
<tr>
<td>FaceFour</td>
<td>78.40</td>
<td>83.00</td>
<td>94.32</td>
<td>94.32</td>
<td>90.91</td>
<td><strong>95.45</strong></td>
<td>88.64</td>
</tr>
<tr>
<td>FacesUCR</td>
<td>76.90</td>
<td>90.49</td>
<td>94.15</td>
<td>78.88</td>
<td>48.00</td>
<td><strong>95.80</strong></td>
<td>89.32</td>
</tr>
<tr>
<td>Fish</td>
<td>78.30</td>
<td>83.30</td>
<td>87.43</td>
<td>95.43</td>
<td><strong>96.00</strong></td>
<td>76.00</td>
<td>93.71</td>
</tr>
<tr>
<td>Gun Point</td>
<td>91.30</td>
<td>90.70</td>
<td><strong>100.00</strong></td>
<td><strong>100.00</strong></td>
<td><strong>100.00</strong></td>
<td>98.67</td>
<td>98.00</td>
</tr>
<tr>
<td>Haptics</td>
<td>37.00</td>
<td>47.70</td>
<td>38.64</td>
<td>43.83</td>
<td>44.16</td>
<td>41.23</td>
<td>43.18</td>
</tr>
<tr>
<td>InlineSkate</td>
<td>34.20</td>
<td>38.40</td>
<td>32.00</td>
<td>44.00</td>
<td><strong>45.45</strong></td>
<td>35.45</td>
<td>42.73</td>
</tr>
<tr>
<td>ItalyPowerDemand</td>
<td><strong>95.50</strong></td>
<td>95.00</td>
<td>84.26</td>
<td>82.41</td>
<td>69.48</td>
<td>83.77</td>
<td>87.56</td>
</tr>
<tr>
<td>Lighting2</td>
<td>75.40</td>
<td><strong>86.00</strong></td>
<td>75.41</td>
<td>67.21</td>
<td>57.38</td>
<td>81.97</td>
<td>70.49</td>
</tr>
<tr>
<td>Lighting7</td>
<td>57.50</td>
<td><strong>72.60</strong></td>
<td>64.38</td>
<td>21.92</td>
<td>17.81</td>
<td>69.86</td>
<td>42.47</td>
</tr>
<tr>
<td>MedicalImages</td>
<td>68.40</td>
<td><strong>73.70</strong></td>
<td>71.05</td>
<td>61.71</td>
<td>59.60</td>
<td>71.97</td>
<td>68.29</td>
</tr>
<tr>
<td>Motes</td>
<td><strong>87.90</strong></td>
<td>93.50</td>
<td>79.71</td>
<td>65.58</td>
<td>??</td>
<td>82.59</td>
<td>68.37</td>
</tr>
<tr>
<td>OliveOil</td>
<td><strong>86.70</strong></td>
<td><strong>86.70</strong></td>
<td>83.33</td>
<td>76.67</td>
<td>83.33</td>
<td>73.33</td>
<td>83.33</td>
</tr>
<tr>
<td>OSULeaf</td>
<td>51.70</td>
<td>59.10</td>
<td>64.46</td>
<td>83.06</td>
<td><strong>83.88</strong></td>
<td>65.29</td>
<td>80.17</td>
</tr>
<tr>
<td>RobotSurface</td>
<td>69.50</td>
<td>72.50</td>
<td><strong>79.70</strong></td>
<td>70.55</td>
<td>66.29</td>
<td>79.70</td>
<td>73.64</td>
</tr>
<tr>
<td>RobotSurfaceII</td>
<td><strong>85.90</strong></td>
<td>83.10</td>
<td>84.26</td>
<td>78.49</td>
<td>51.31</td>
<td>84.47</td>
<td>84.26</td>
</tr>
<tr>
<td>SwedishLeaf</td>
<td>78.70</td>
<td>79.00</td>
<td><strong>90.24</strong></td>
<td>89.64</td>
<td>66.24</td>
<td>88.80</td>
<td>88.88</td>
</tr>
<tr>
<td>Symbols</td>
<td>90.00</td>
<td>95.00</td>
<td><strong>90.45</strong></td>
<td><strong>96.38</strong></td>
<td>96.08</td>
<td>90.05</td>
<td>96.18</td>
</tr>
<tr>
<td>WordsSynonyms</td>
<td>61.80</td>
<td>64.90</td>
<td>72.41</td>
<td>71.00</td>
<td>62.54</td>
<td><strong>73.35</strong></td>
<td>67.24</td>
</tr>
</tbody>
</table>

Table 1: Accuracy rates for each distance measure. Following the notation in [10], the symbols ▼, ▲ and • represent data sets generated from figure shapes, human movements and all remaining data sets, respectively.

For the CRPCD approach with the MPEG-1 video encoding algorithm we used the test setup described in [10] but implemented our new compression distance approach to preclude distorted results due to different configurations of the video encoding algorithms.

In the evaluation we used the nearest neighbor method to classify the data sets provided by the UCR Time Series Classification/Clustering Page [5]. Due to the fact that we had four different distance measures to test, two of which were based on the CRPCD
method that requires 8 video compressions to be made, we removed the largest data sets from the testing corpus. The results of the evaluation can be seen in Table 1.

![Figure 3](image.png)

**Fig. 3:** A plot comparing the accuracy of the RPCD method using MPEG-1 (bottom-right) and MPEG-2 (top-left) video encoding algorithms. Each point represents a data set. In all data sets in the white area of the plot the MPEG-2 approach outperformed MPEG-1. Following the notation in [10], the symbols \( \triangledown \), \( \triangle \) and \( \bullet \) represent data sets generated from figure shapes, human movements and all remaining data sets, respectively.

### 4.2 Results and Discussion

As seen in Table 1, each tested setup, except for the CRPCD-MPEG2 approach, had roughly the same number of data sets where it scored the best result (values in bold). This shows that the different measurement approaches complement each other in a way that none of the tested measures outperform all the others.

Notably large time series like InlineSkate (1882 data points) or Haptics (1092 data points) performed better with the newer video compression algorithms. We believe that this is due to the fact that with large plots the overhead and the intra-frame encoding are less significant. Thus the resulting file size depends more on the inter-frame compression of two images.

The approach to calculate the RPCD measure with the newer video encoding algorithm MPEG-2 showed no significant improvements to the RPCD implementation as can be seen in Figure 3. Notably, most of the data sets that were generated by human movement performed significantly worse. Because the MPEG-2 algorithm is used as a black box it is difficult to identify the reasons of this unexpected result. Possible reasons may be due to the fact that the MPEG-2 video encoding produces a bigger file overhead and that the intra-frame compression improved over the MPEG-1 algorithm. This may lower the significance of the inter-frame compression in the video file.
Fig. 4: A plot comparing the accuracy of the RPCD measure to the CRPCD measure - both with the MPEG-1 video encoding algorithm - where each point represents a data set. In all data sets in the white area of the plot CRPCD outperformed RPCD. Following the notation in [10], the symbols ▼, ▲ and ■ represent data sets generated from figure shapes, human movements and all remaining data sets, respectively.

Fig. 5: A plot comparing the accuracy of the RPCD measure with MPEG-1 video encoding to the CRPCD measure with MPEG-2 video encoding, where each point represents a data set. In all data sets in the white area of the plot CRPCD-MPEG2 outperformed RPCD-MPEG1. Following the notation in [10], the symbols ▼, ▲ and ■ represent data sets generated from figure shapes, human movements and all remaining data sets, respectively.
The CRPCD method tested with the MPEG-1 algorithm shows slight improvements to the original RPCD approach. Especially time series that were generated by human movement improved in contrast to RPCD as can be seen in Figure 4. However it is notably that the calculation time of the CK-2 algorithm is significantly higher than the time of the RPCD method because our approach requires two times as many plots to be generated and videos to be encoded.

We tested the newer video encoding algorithm MPEG-2 with the CRPCD approach combined, to see how these two factors influence each other. Apart from a few outliers, the MPEG-2 CRPCD method performs worse than the MPEG-1 CRPCD approach, as can be seen in Figure 5. We reason that in line with our hypothesis that the MPEG-2 algorithm undermines the inter-frame compression with the intra-frame compression, the CRPCD method further lessens the significance of the compression between two plots.

Additionally to the MPEG-2 video encoding algorithm we also tested the more advanced MPEG-4 algorithms. Overall the data sets that performed well with the MPEG-2 algorithm also performed well with the MPEG-4 video encoder and data sets that had less accuracy with MPEG-2 also had less accuracy with MPEG-4.

5 Conclusion

In this work we evaluated the use of more advanced video encoding algorithms like MPEG-2 and MPEG-4 with the RPCD measure. We furthermore introduced a new recurrence plot-based distance measure CRPCD that utilizes cross recurrence plots.

We showed that more advanced compression algorithms do not necessarily result in higher classification accuracy. However, we observed that the MPEG-2 and MPEG-4 algorithms yield better performance for relatively long time series. Though none of the classifiers is consistently better, the results show that they complement one another.

In future work we intend to evaluate the correlation between the size of a time series and the accuracy of video compression-based distance measures. Future work may also include the modification of video encoding algorithms in a way that the compression only takes place in temporal space so that overhead and unwanted intra-frame-compression get minimized.
Bibliography


