Bachelor Thesis

PCCA+ and its Application
to Spatial Time Series Clustering

Free University of Berlin
Department of Mathematics and Computer Science

Alexander Sikorski
Supervisor: PD Dr. Marcus Weber

March 18th, 2015

Declaration

I hereby declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institute. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Alexander Sikorski
Berlin, March 18th, 2015

Contents

1 Introduction

In this thesis, we develop an algorithm for clustering spatial time series into a prescribed number of clusters based on their spatial and dynamical properties.

After an introduction to the underlying theoretical background we will review known results about the *Robust Perron Cluster Cluster Analysis* (PCCA+), which forms the basis for our application. PCCA+ allows to identify metastable clusters in Markov chains which are configurations of the system which are likely to persist for a longer time. In the course, we will extend the known results by a stochastic interpretation for the propagator matrix which encodes the time evolution on the clusters.

We then explain a method to turn spatial time series into a Markov chain to obtain a spatial clustering by further application of PCCA+, respecting the dynamic information. Finally, we will apply that method to data obtained by tracking human eye fixations while these look at different paintings to detect the depicted objects. This can be seen as a form of object recognition which does not rely on the image data itself but detects the objects based on the human recognition reflected in their eye movement.

The presented program was developed in cooperation with the Zuse Institute Berlin and the University of Potsdam and I would like to thank Dr. Weber and Prof. Dr. Kliegl for their support.

2 Theoretical background

2.1 Introduction to Markov chains

Let S be any finite set, i.e. $S := \{s_1, ..., s_N\}$. A *Markov chain* on S is a stochastic process, consisting of a sequence of random variables $X_i : \Omega \to S$, $i \in \mathbb{N}$ satisfying the Markov property:

$$P(X_{t+1} = x | X_1 = x_1, X_2 = x_2, ..., X_t = x_t) = P(X_{t+1} = x | X_t = x_t) \, \forall t \in \mathbb{N}.$$

It is common to interpret S as the state space of possible outcomes of measurements at time step t represented by X_t. The Markov property assures that the transition probabilities to the next time step x_{t+1} only depend on the current state x_t. This means that the process at time t has no memory of its previous history $(x_1, ..., x_{t-1})$, sometimes this is also called the memoryless property.

We will furthermore assume that the process is autonomous, i.e. not explicitly depending on the time:

$$P(X_{t+1} = x | X_t = y) = P(X_t = x | X_{t-1} = y) \forall t \in \mathbb{N}.$$

This does not really impose a restriction since any non-autonomous process can be turned

into an autonomous one: By adding all possible times to the state space S using the Cartesian product $S' := \mathbb{N} \times S$, the explicit time-dependence of the process on S can be implicitly subsumed by an autonomous process on S'.

Since S is finite, we can, enumerating all states in S, encode the whole process in the right stochastic *transition matrix*

$$P_{ij} := P\left(X_{t+1} = j | X_t = i\right),$$

in which case right stochastic means that each row has row sum one and propagation of states is realized by right application of P.

A *stationary distribution* is a row vector π, satisfying

$$\pi P = \pi, \sum_{i=1}^{N} \pi_i = 1.$$

Given a stationary distribution π, we denote by D_π the diagonal matrix with π on its diagonal.

Although we only consider a discrete state space in this thesis, the results are extendible to continuous state spaces as well. A natural way is using a set-based discretization dividing the state space into a finite mesh of subsets. For high dimensional state spaces, as for example met in molecular dynamics, this approach exhibits the curse of dimensionality, as the size of the mesh grows exponentially with the dimensions. Weber developed a meshless version of PCCA+ using a global Galerkin discretization[9] as solution to this problem .

2.2 Clustering of the state space

The goal of PCCA+ is to reduce the complexity of analysis of the Markov chain by a dimension reduction of the state space. To formalize this we will now introduce the concept of clustering, which is subsuming different states of the state space to a smaller set of $n \in \mathbb{N}$ clusters $C := \{1, ..., n\}$.

The simplest possibility is assigning each state $k \in S$ to a cluster $i \in C$ which can be encoded by means of the *characteristic vector* $\chi_i \in \{0,1\}^N$:

$$\chi_{i,k} = \begin{cases} 1, & \text{if state } k \text{ belongs to cluster } i \\ 0, & \text{else} \end{cases}.$$

Due to its discrete nature, this *crisp clustering* approach, used by the *Perron Cluster Cluster Analysis* (PCCA) [1], has the disadvantage of not being robust against small perturbations since continuous changes in P finally result in discontinuous changes in the clustering.

Deuflhard and Weber therefore developed a robust version, *Robust Perron Cluster Analysis* (PCCA+) [2], by making use of a *fuzzy clustering* representing each cluster by an *almost*

characteristic vector

$$\chi_i \in [0,1]^n \,. \tag{2.1}$$

Almost characteristic vectors $\{\chi_i\}_{i=1}^n$ satisfying the partition of unity property

$$\sum_{i=1}^n \chi_i = 1 \tag{2.2}$$

are called *membership vectors* as they describe the relative membership of each state to each cluster. We will refer to the matrix collection $\chi := (\chi_i)_{i=1}^n \in \mathbb{R}^{N \times n}$ of the *membership vectors* as a *clustering*, whereas in the field of computational chemistry it is also referred to as *conformations*.

2.3 Galerkin projection of the transition matrix

The coupling Matrix

To represent the dynamics on the reduced/clustered state space in the case of a *crisp clustering* χ, i.e. $\chi_i \in \{0,1\}$, Deuflhard et al. [1] introduced the *coupling matrix*

$$W_{ij} := \frac{\langle \chi_j, P\chi_i \rangle_\pi}{\langle \chi_i, 1 \rangle_\pi} = \frac{\chi_j^T D_\pi P \chi_i}{\pi^T \chi_i},$$

or in matrix notation

$$W := \text{diag}\left(\chi^T \pi\right)^{-1} \chi^T D_\pi P \chi.$$

The entries W_{ij} can thus be interpreted as conditional transition probabilities from cluster i to cluster j, given the starting distribution π.

In the *fuzzy clustering* setting, the problem arises that it is not clear anymore which state belongs to which cluster. It is therefore convenient to interpret the membership of state j to cluster χ_i, χ_{ij}, as the probability of measuring state j belonging to cluster χ_i. Then, W_{ij} denotes the expectation value for measuring cluster χ_j after propagating the density given by χ_i.

Note, however, that even if no real transitions are actually happening in the state space, we still may count transitions between clusters since we measure the same state once belonging to one and then to another cluster, as demonstrated in example 3.

One of the main motivations for developing PCCA+ was the wish to identify so called *metastable conformations* of molecular systems, e.g. to analyze the effectivity of active pharmaceutical ingredients in Computational Molecular Design (for a overview over this approach see [7]). These *conformations* are *almost invariant aggregates* of states, i.e. *membership vectors* with high self-transition probabilities, guaranteeing that the system resides in these states on longer timescales.

This can be formalized, as proposed by Huisinga [3], by the definition of the *metastabil-*

ity of *membership vectors* as the trace of the corresponding *coupling matrix*: $\operatorname{tr}(W)$. Note, this does not need to correspond with a high probability of the cluster.

The propagator matrix

Unfortunately, the projection via the *coupling matrix* does not commute with time propagation and therefore cannot be used for long term analyses of the underlying Markov chain.

As remedy, Kube and Weber [5] proposed the *coarse propagator matrix*

$$P_C := \left(\chi^T D_\pi \chi\right)^{-1} \chi^T D_\pi P \chi \tag{2.3}$$

which coincides with the *coupling matrix* W in the *crisp clustering* setting. Assuming that χ is a linear combination of vectors spanning a P-invariant subspace satisfying an invertibility condition, it has the advantage that discretization via χ and time propagation commute, i.e.

$$P\chi = \chi P_C. \tag{2.4}$$

This property ensures that the *coupling matrix* represents the right dynamics of the underlying Markov chain on the reduced state space, even for iterative application, i.e. $P^n \chi = \chi P_C^n$.

Theorem 1. *Let $\chi = XA$, $X \in \mathbb{R}^{N \times n}$, $A \in \mathbb{R}^{n \times n}$ satisfying the subspace condition*

$$PX = X\Lambda \tag{2.5}$$

for some $\Lambda \in \mathbb{R}^{n \times n}$ and $C := X^T D_\pi X$ be invertible.

Then the P_C is conjugate to Λ and discretization-propagation commutativity *(2.4) holds*.

Proof. We calculate

$$
\begin{aligned}
P_C &= \left(\chi^T D_\pi \chi\right)^{-1} \chi^T D_\pi P \chi \\
&= \left(A^T C A\right)^{-1} A^T C \Lambda A \\
&= A^{-1} C^{-1} A^{-T} A^T C \Lambda A \\
&= A^{-1} \Lambda A,
\end{aligned}
\tag{2.6}
$$

which implies

$$P\chi = PXA = X\Lambda A = XAA^{-1}\Lambda A = \chi P_C.$$

$\square$

This form of the theorem constitutes a small generalization towards its so far published versions in which $C = Id$ was assumed instead of invertibility.

Stochastic interpretation

Due to the matrix inversion, the *propagator matrix* can have negative entries as shown in example 3 below, thus prohibiting a natural stochastic interpretation.

We will therefore shed some light into the connection between the *coupling-* and the *propagator matrix* making use of the notation of the *restriction-* and *interpolation operators* introduced by Kube and Weber [5]:

$$
\begin{aligned}
R : \mathbb{R}^N &\rightarrow \mathbb{R}^n, \, x \mapsto x\chi \\
I : \mathbb{R}^n &\rightarrow \mathbb{R}^N, \, x \mapsto x D_{\tilde{\pi}}^{-1} \chi^T D_\pi
\end{aligned}
$$

with $\tilde{\pi} = \pi R$ and χ as above, where we apply them from the right in line with the used notation of right-stochastic matrices.

These provide the transformations between the (fine-grained) configuration space and the (coarse-grained) cluster space. These are in the sense that $IRw = w$, i.e. I reconstructs the fine-grained density lost by the restriction R using the fine-grained stationary density.

This allows to reformulate the *coupling-* and *propagator matrix as*

$$
\begin{aligned}
W &= IPR \\
P_C &= (IR)^{-1} IPR.
\end{aligned}
$$

Now, consider the situation when setting $P = Id$ with a fuzzy clustering. Then $W = IR = \tilde{D}_\pi^{-1} \chi^T D_\pi \chi \neq Id$ as different clusters overlap. The result corresponds to the transitions which are introduced to the coarser system due to the overlap.

As this overlap would be applied on every iteration of W, it would lead to increased mixing between the states resulting in wrong long-term results. P_C grants the desired commutativity (2.4) by factoring out these transitions.

This also shows how we can compute a corresponding stochastically interpretable *coupling matrix* for larger times corresponding to n iterations, W_n, from the smaller matrix P_c:

$$
W_n := IP^n R = IRP_C^n.
$$

2.4 Example Processes

To demonstrate the connections between the different projections we now will show some example systems.

Example 1: The decoupled system

Consider

$$P = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

In this ideal decoupled system we have two invariant subspaces spanned by the so called *Perron eigenvectors* with eigenvalue 1, the vectors

$$\begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \text{ and } \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

These can be interpreted as a *crisp clustering* and, assuming an equidistributed starting distribution, we can compute the, in the *crisp* case coinciding, matrices

$$W = P_C = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Example 2: The 3-pot

Next, we will consider the stationary Markov chain on three states with a *fuzzy clustering*:

$$P := \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \chi = \begin{pmatrix} 1 & 0 \\ \frac{1}{2} & \frac{1}{2} \\ 0 & 1 \end{pmatrix}, \pi = \frac{1}{3}\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

According to our definitions we now compute

$$P_C := \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, W = \frac{1}{6}\begin{pmatrix} 5 & 1 \\ 1 & 5 \end{pmatrix}.$$

We thus observe that P_C contains the expected stationary dynamics on the reduced state space while W accounts for the possible transitions of observing the second state once in cluster 1 and once in cluster 2, due to the overlap in the *clustering*.

Example 3: Negative entries

Let us now consider

$$P = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \chi = \begin{pmatrix} 1 & 0 \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}, \pi = \frac{1}{2} \begin{pmatrix} 1 & 1 \end{pmatrix}.$$

Computation of the propagator matrix now leads to negative entries:

$$P_C = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ 3 & -1 \end{pmatrix}.$$

Let us first compute the propagation of the normalized density corresponding to χ_2:

$$(0,1) \cdot P = (1,0),$$

i.e. s_2 is propagated to s_1. The corresponding computation on the cluster space is

$$(0,1) \cdot P_C = \frac{1}{2} (3,-1).$$

Here, the negative entry amounts for the overlap of the clusters and is necessary to encode state s_1: As both clusters hold an amount of s_2 we use a linear combination to eliminate this. We thus may interpret P_C acting to the basis of clusters eliminating the overlap.

We can calculate the corresponding density on the state space by applying the interpolation operator

$$I = \begin{pmatrix} \frac{2}{3} & \frac{1}{3} \\ 0 & 1 \end{pmatrix}, \frac{1}{2} (3,-1) \cdot I = (1,0).$$

3 PCCA+

In this section, we will construct the *Robust Perron Cluster Cluster Analysis* algorithm, introduced in [2]. We first will construct the matrix X spanning the required invariant subspace and examine the possible linear transformations A mapping these to a set of membership vectors. Finally, we propose different objectives to an optimization problem to specify a "good" solution.

Unlike previous treatises, we will not restrict ourselves to reversible processes and provide a more general version of PCCA+ using the Schur decomposition thus reflecting the newest developments in [10].

Note, that we impose a fixed cluster number n. An overview over methods for estimating the cluster number based on different criteria is given by Röblitz, Weber[8].

PCCA+ will construct the clusters described by the *membership vectors* as a linear combination of eigenvectors. This guarantees that χ spans an invariant subspace, thus leading to preservation of the slow time-scales. By choosing the $n < N$ eigenvectors with the

largest eigenvalues, one hopes to preserve the principal dynamics of P. A good indicator towards this goal is that eigenvectors with large eigenvalues represent conformations with a high degree of self-mapping. Deuflhard et al. [1] have furthermore shown that the desired metastability is bounded from above by the sum of the chosen eigenvalues and for ϵ-perturbations of the coupling of uncoupled Markov chains also from below by $\sum \lambda_i - O\left(\epsilon^2\right)$, justifying the choice of large eigenvalues.

3.1 Subspace construction

For the construction of the invariant subspace, we make use of the *real Schur decomposition* decomposing a matrix $P = Q\Lambda Q^{-1}$ into a orthonormal matrix Q, whose columns are called the *Schur vectors,* and an upper quasi-triangular (1-by-1 and 2-by-2 blocks on its diagonal) matrix Λ, called the *Schur form*. The columns of Q are called the Schur vectors of P. The eigenvalues of P appear on the diagonal of Λ, where complex conjugate eigenvalues correspond to the 2-by-2 blocks.

Compute the Schur decomposition of $\tilde{P} = D_\pi^{\frac{1}{2}} P D_\pi^{-\frac{1}{2}}$. Then reorder the *Schur form* so that the upper left $n \times n$ block contains the n largest eigenvalues, recomputing the corresponding *Schur vectors* using a Schur reordering algorithm [4]. Note, that in the case of complex conjugate eigenvalues we have to select or discard the whole 2-by-2 blocks. Let us denote the resulting *Schur form* and *-vectors* by Λ, $\tilde{X}$. Then, defining $X = D_\pi^{-\frac{1}{2}} \tilde{X}$ and inserting into $\tilde{P}\tilde{X} = \tilde{X}\Lambda$ leads to $PX = X\Lambda$. Furthermore, due to orthonormality of $\tilde{X}$, we have $X^T D_\pi X = Id$. So X and Λ satisfy the conditions for Theorem 1.

3.2 The feasible transformation set

Given the invariant subspace X, we will now examine the set of feasible matrices $F_A \subset \mathbb{R}^{n \times n}$ for the transformation A, leading to actual *membership vectors* $\chi := XA$. As P is stochastic, the vector $e = (1, ..., 1)^T$ is mapped to itself and thus forms an eigenvector to eigenvalue one which for stochastic matrices also is the largest eigenvalue. By the reordering of X and due to $X^T D_\pi X$, we have $X_{i,1} = 1$, $i = 1, ..., N$. Thus, one can reformulate the *positivity* (2.1) and *partition of unity* (2.2) conditions in terms of the matrices X and A, leading to the following constraints for A:

$$A_{1,j} \;\geq\; -\sum_{k=2}^{n} X_{ik} A_{kj}, \, i = 1, ..., N, \, j = 1, ..., n \text{ (positivity)}, \qquad (3.1)$$

$$A_{i,1} \;=\; \delta_{i,1} - \sum_{j=2}^{n} A_{ij}, \, i = 1, ..., n \text{ (partition of unity)} \qquad (3.2)$$

Since these constraints are linear in A, the set F_A is a convex polytope.
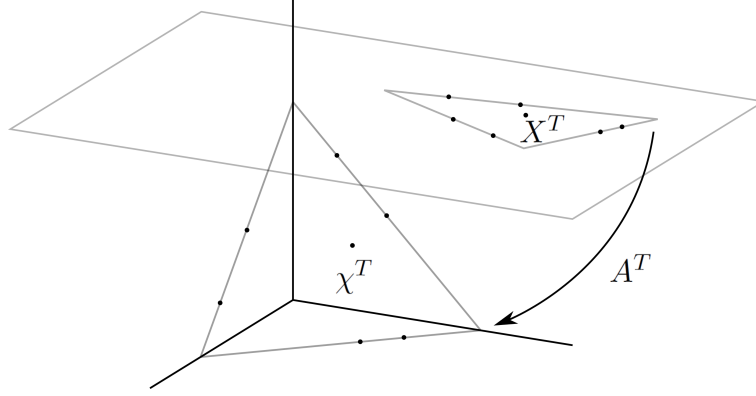
Figure 3.1: Schematic illustration of the linear transformation mapping the row vectors of the eigenvectors X (points on the $z = 1$ hyperplane) onto the standard 2-simplex.

Geometrically, one might think of the N rows of the matrix χ as points in the space $\mathbb{R}^n$ (see Figure 3.1). The *positivity* (2.1) and *partition of unity* (2.2) conditions force these points to lie on the standard $(n-1)$-simplex Δ. Now, if $\chi^T = A^T X^T$ this means that the matrix A maps the N rows of the eigenvector matrix X to that simplex. As we have seen the first component of each row is 1, thus, all rows lie on the hyperplane with first component 1. They furthermore are contained in a bounded region, and thus we can map them linearly onto Δ via a linear map A.

Thus, we directly see that the set F_A is not empty. It furthermore is infinite, as we can always shrink the image further and it will still fit into Δ.

3.3 Optimization

As we have the choice between infinitely many possible solutions for $A \subset F_A$, we will now specify and motivate different optimization objectives to choose a specific solution by the optimization problem.

Maximal scaling condition

Assuming (*maximality assumption*), that the convex hull $\mathrm{co}\,(X)$ of the rows of X already has the form of an $(n-1)$-simplex, we can now choose A uniquely (up to permutation) to map this exactly onto Δ, which among all the ways of mapping X into Δ gives us the highest distinctiveness between the resulting clusters. This assumption is equivalent to the situation that for each corner there exists a row getting mapped into that corner, i.e.

$$\max_{i=1..N} \chi_{ij} = 1, \; j = 1, ..., n,$$

justifying its name.

As in general the *maximality assumption* is not met, it seems natural to turn it into an

optimization problem. This has been done in [2, 9] by imposing maximization of the *maximal scaling condition*

$$I_1\left(A\right) := \sum_{j=1}^{n} \max_{i=1..N} \chi_{ij} \leq n_C.$$

Assuming that the *maximality assumption* is almost met, i.e. $\max_{i=1..N} \chi_{ij} \approx 1$, $j = 1, ..., n$, Weber [9] proposes to determine the maximizing indices by the *index mapping algorithm* (Algorithm 2), turning this convex optimization problem into a linear one:

$$I_1\left(A\right) = \sum_{i,j=1}^{n} X_{ind(X)_j,i} A_{ij}$$

In [9], Weber furthermore shows that $W_{jj} \leq \max_{i=1..N} \chi_{ij}$, which implies that I_1 is an upper bound for the metastability, which thus should be large.

Note, that this objective, ignoring the data points not being the maxima, cannot distinguish between differences in the interior of the convex hull, leading to possibly non-optimal transformation matrices A as illustrated in Figure 3.2.
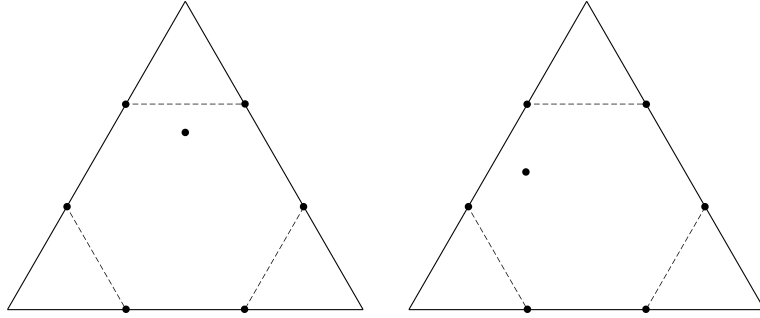

Figure 3.2: Mapping of 7 rows of the eigenvectors (affine hexagon with an interior point) to a 2-simplex. While I_1 cannot differentiate between the two mappings, I_2 will choose the second as it provides a crisper assignment of the interior point.

Maximal metastability condition

Another choice might be optimizing directly towards a maximal metastability as done by Deuflhard and Weber [2, 9]

$$I_2\left(A\right) := \mathrm{trace}\left(W\right) = \sum_{i=1}^{n} \lambda_i \sum_{j=1}^{n} \frac{A_{ij}^2}{A_{1,j}}$$

where they establish the latter equation making use of $\pi_i = A_{1,i}$ ([9], Lemma 3.6).

Crispness objective

Röblitz [8] argues that the stochastic interpretation of W is not valid in the fuzzy setting, due to the overlap. Optimization of the trace of P_C makes no sense as it is similar to Λ, see (2.6), and therefore independent of A. She therefore suggests maximization of

$$I_3 := \text{trace}\,(IR) = \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{A_{ij}^2}{A_{1,j}}$$

which is similar to the *maximal metastability condition* with the corresponding P replaced by the identity.

Maximizing the trace minimizes the off-diagonal entries of IR leading to the least amount of clustering-induced transitions and therefore to a as crisp as possible clustering.

Unconstrained Optimization

Due to the high number of inequality constraints (3.1), solving these linear or convex problems may still be very time consuming. Following Deuflhard and Weber [2, 9], we will now show how to turn this constrained into an unconstrained optimization problem by enforcing the constraints after each iteration.

Define the set F_A' by the equality constraints

$$
\begin{aligned}
A_{i,1} &= \delta_{i,1} - \sum_{j=2}^{n} A_{ij}, \ i = 1, ..., n \\
A_{1,j} &= -\min_{l=1,...,N} \sum_{i=2}^{n} X_{li} A_{ij}, \ j = 1, ..., n.
\end{aligned}
\tag{3.3}
$$

Comparing these equalities to (3.1), one easily checks that $F_A' \subset F_A$.

Now consider the following *feasibilization algorithm* $F : \mathbb{R}^{(n-1)\times(n-1)} \twoheadrightarrow F_A'$, mapping any arbitrary matrix $\left(\tilde{A}_{ij}\right)_{i,j=2,...,n}$ to a feasible transformation matrix A and thus enforcing the desired constraints:

Algorithm 1: Feasibilization algorithm

1. For $i = 2, ..., n$ define $\tilde{A}_{i,1} := -\sum_{j=2}^{n} \tilde{A}_{ij}$

2. For $j = 1, ..., n$ define $\tilde{A}_{1,j} := -\min_{l=1,...,N} \sum_{i=2}^{n} X_{li} \tilde{A}_{ij}$

3. For $i, j = 1, ..., n$ define $A_{ij} := \frac{\tilde{A}_{ij}}{\sum_{j=1}^{k} \tilde{A}_{1,j}}$

4. Return A

Steps 1 and 2 guarantee feasibility of $\tilde{A}$ with respect to (3.3) for $i = 2, ..., n$ respectively $j = 1, ..., n$. As these equalities are linear in A, they are invariant under scalar multipli-

cation and step 3 now furthermore assures the equality (3.3) for $i = 1$. Thus, F indeed maps to F'_A. Furthermore, taking any matrix $A \in F'_A$, dropping the first row and column to get $\tilde{A}$ and computing $F\left(\tilde{A}\right) = A$, we see that F is surjective.

As any objective function I_i, $i = 1, 2, 3$ is convex over F_A it attains its maximum at one of the vertices $v(F_A)$, which are contained in F'_A (for a proof see [9], Lemma 3.5). Thus, we can also optimize the function $F \circ I_i$ over $\mathbb{R}^{(n-1) \times (n-1)}$ and have thereby transformed the constrained optimization problem in n^2 unknowns to an unconstrained in $(n-1)^2$ unknowns.

As the *feasibilization algorithm* is not differentiable, Deuflhard and Weber [2] propose the use of the nonlinear simplex method of Nelder and Mead [6] as local optimization routine.

Initial guess

Based on Weber and Galliat [11], we outline the *inner simplex algorithm* determining an initial guess for the matrix A by constructing a simplex surrounding all row-points and then computing the transformation to the standard simplex, thus, turning the global into a local optimization problem.

The first step, the *index mapping algorithm,* searches for the indices i_j of the successively farthest linear independent rows. It starts by choosing the largest row vector as starting point and then iteratively adds the points with the largest distance to the hyperplane spanned by the chosen points so far:

Algorithm 2: Index mapping algorithm

1. Find starting point: $i_1 := \text{argmax}_{j \in C} \|X_{\cdot, j}\|_2$

2. Translate to origin: For $i \in S$ set $X_{i, \cdot} \hookleftarrow X_{i, \cdot} - X_{i_1, \cdot}$

3. For $j = 2, ..., n$

   a) Find index of the farthest point: $i_j := \text{argmax}_{j \in C} \|X_{\cdot, j}\|_2$

   b) Projection to hyperplane by Gram-Schmidt process: $X \hookleftarrow X - \dfrac{XX_{i_j, \cdot}^T \otimes X_{i_j, \cdot}}{\left\|X_{i_j, \cdot}\right\|_2}$

4. Return the indices i_j

The *inner simplex algorithm* uses these indices to construct of the n extremal points to construct the matrix A mapping these to the vertices of Δ:

Algorithm 3: Inner simplex algorithm

Return $A(X) := (X_{ij})^{-1}_{i=i_1, ..., i_n, \, j=1, ..., n}$ with $i_1, ..., i_n$ computed by the *index mapping algorithm*

In the case of the *maximality assumption*, X spans a $(n-1)$-simplex, and the *index mapping algorithm* determines its vertices, thus $\text{co}\,(AX) = \Delta$ and $A \in v\,(F_A)$ maximizes I_1. For the general case though, Weber [9] (Lemma 3.13, Theorem 3.14) has shown that the following statements are equivalent:

1. The convex hull $\text{co}\,(X)$ of X is a simplex.

2. The result of the *inner simplex algorithm* is feasible, i.e. $A \in F_A$.

3. $A \in v\,(F_A)$ and therefore maximizes I_1.

Therefore, the result is not feasible in the generic case. If, however, the *maximality assumption* almost holds, i.e. the convex hull of X is a small perturbation of a simplex, which according to Weber [9] (3.4.4) is satisfied in many applications, the algorithm still gives a solution near the unperturbed solution. Thus, A is near a vertex of the set F_A and thus a good initial guess for a local optimization of the unconstrained optimization.

3.4 The PCCA+ algorithm

Once we have decided on an objective function we can now put together all steps:

Algorithm 4: PCCA+

1. Given P and π, compute X, Λ using the weighted Schur decomposition as in section 3.1

2. Determine the, in general infeasible, initial guess $A_0 := A\,(X)$ by the *inner simplex algorithm*.

3. Perform an iterative local optimization A_0, $A_1, ...$ of the objective function I_1, I_2 or I_3. In each step $A_k \to A_{k+1}$ only update the elements $A_{k,ij}$, $i, j \neq 1$ without constraints, then use Algorithm *1* to get a feasible matrix A_k before evaluating the corresponding objective function.

4. Return X and A.

Extension to time-continuous Markov chains

PCCA+ is also applicable to the clustering of time-continuous Markov chains (c.f. [5]). In that case, the *transition matrix* P gets replaced by a *transition rate matrix* Q having row-sum zero and non-negative off-diagonal entries. The transitions after a fixed time can then be computed by

$$P(t) = e^{tQ}.$$

In that case, the eigenvectors of P and Q are the same and the eigenvalues of P are the exponential of the corresponding eigenvalues of Q. As the exponential is monotone, the

eigenvectors with highest absolute eigenvalue, near 1, of P correspond to the eigenvectors with smallest absolute value, near zero, of Q.

So by selecting the eigenvectors with smallest eigenvalue of Q we can compute the corresponding clustering for time-continuous Markov chains.

4 Application to eye tracking data

This algorithm was applied to experimental eye tracking data obtained by the Department of Psychology of the University of Potsdam with the goal to detect objects as metastable clusters using just the dynamics of the human eye, i.e. without any data of the image itself, and thus provides a way of interpreting the humans object recognition expressed through the eye movements.

4.1 The experiment and model

A group of test persons was presented different pictures for about 10 seconds, during which an eye tracker measured their fixations $f_i \in \mathbb{R}^2$ and their respective durations $t_i \in \mathbb{R}$. For subsequent analysis it is necessary to group different areas of the image into *areas of interest (AOE)* which correspond to subjectively identified objects in the corresponding picture.

To apply PCCA+, we need to turn this spatial time series into a Markov chain.

We model each fixation as a *random choice* on a spatial grid, weighted by a Gaussian of the distance to the grid points, and then construct a *Markov chain* by counting the induced transitions on the grid points. Assuming, that humans, when looking at the pictures, do not jump randomly between all recognized objects but remain for some fixations inside one *AOE*, this behaviour should recur as high metastability of a clustering, corresponding to the *AOEs.*

4.2 Implementation

As state space we choose a spatial grid $S := \{s_i\}$, where the natural choice is using all fixation coordinates as grid ($s_i = f_i$) or alternatively use some spatial clustering algorithm (e.g. k-means) to reduce the computational effort of the following PCCA+ routine.

Introducing a parameter σ, we assign a membership of each fixation to each grid point weighted by a Gaussian of the distance between them, i.e. for each fixation f_i and each state s_j:

$$M_{ij} := \frac{e^{\frac{\left|f_i - s_j\right|^2}{2\sigma^2}}}{\sum_j e^{\frac{\left|f_i - s_j\right|^2}{2\sigma^2}}} \tag{4.1}$$

This *mass matrix* assures that nearby fixations "overlap", adopting the metric information contained in the fixation data to the Markov process. Thus, the parameter σ, scaling the distance between points, can be interpreted as a spatial coupling constant.

We then choose a fixed time step $\Delta\tau$ as grid size for the time discretization, along which we count the transitions between the states weighted with the corresponding fixation transitions, and row-normalize it to generate a *transition matrix*. In detail, for the transitions from state i to j we have

$$P_{ij} \quad = \quad \frac{\sum_{s=0} M_{f_s,i} M_{f_{s+1},j}}{\sum_{s=0} M_{f_s,i}},$$

where f_s denotes the current fixation at time $s\Delta\tau$.

Once we have constructed P this way, we now compute the stationary distribution, being the left eigenvector to eigenvalue one, and pass them to PCCA+, which in return gives us X and A to compute the clustering $\chi = XA$.

As a final step, we discretize this fuzzy clustering by assigning to each state s_i the cluster c_i with the maximal share:

$$c_i = \text{argmax}_j \chi_{ij}. \tag{4.2}$$

Note, that choosing this discretization of the fuzzy clustering, some clusters may never be assigned, when being dominated by other clusters on every grid point.

In case of preclustering via k-means, the cluster assignment of the grid is passed to the corresponding fixations according to the k-means assignments.

The program was implemented in Julia and its source will be published on https://github.com/kliegl/Hokusai.

4.3 Choice of the parameters

The desired number of clusters, n, was chosen near the number of objects recognized by the experimenter. This, of course, is a subjective choice, but the number of clusters in general depends on the desired resolution of the clustering and thus on the further application. For example, imagine a picture of a bookshelf with books, here one might recognize either the whole shelf, the books, or their titles as objects.

The time step size $\Delta\tau$ should be chosen as large as possible without skipping too many transitions. If it is chosen too large some fixations will be skipped resulting in loss of information and thus leading to a worse clustering. If, on the other hand, chosen too small we count one fixation as multiple self-transitions, thus weakening the effect of the real transitions, favouring the spatial over the dynamic information.

The parameter σ introduces the spatial information and can thus be considered as a weight between dynamic and spatial clustering and is therefore inherently necessary. While small σ values favour the dynamic information, this can lead to scattered clusters

neglecting the spatial component. Large σ values will lead to a more regular and convex clustering by enforcing a stronger spatial coupling between nearby fixations.

For a general routine $\Delta\tau$ should be chosen small enough for most fixations to be counted and then increase σ starting from small values to reach the desired spatial regularity.

The *maximal scaling condition* was used for the following clusterings, as it provided the best results.

4.4 Results

Each of the following clusterings was computed based on about 2000 fixations. Each fixation is marked as a dot, with the color representing the corresponding cluster.
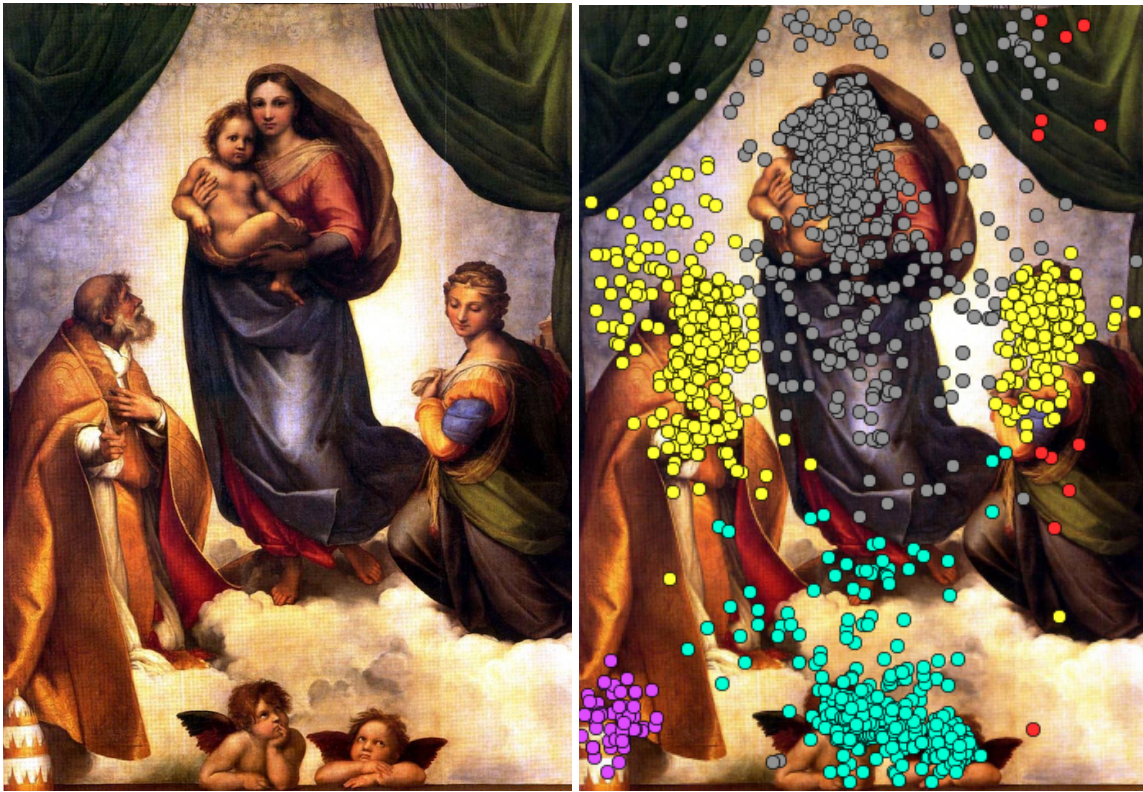
The Sistine Madonna

Figure 4.1(a) depicts Raphael's "Sistine Madonna" oil painting, one of the last of his many Madonnas, commissioned by Pope Julius II for the Monastery of San Sisto in Piacenza, Italy. In the middle, we see the Madonna, holding Jesus, her child. She is flanked by Saint Sixtus and Saint Barbara. To their feet, we can see the two cherubim who became quite popular on their own in the last centuries as motives for postcards, advertisements, gifts and home décor. In the bottom left, we can see the Pope's crown, the Papal tiara.

In (b) we chose a small σ value to emphasize the dynamic share of the clustering. As a consequence Saint Sixtus and Saint Barbara are clustered into one cluster, indicating back and forth movements of the observer, although they are separated by the Madonna. This non-convex clustering is a feature specific to the analysis of the dynamics which can not be reached by purely spatial clustering, e.g. k-means. Furthermore, the Madonna with her child and the Papal tiara are separated. The noisy red cluster is typical for small σ values, as small cycles get uncoupled from the rest of the fixations due to the missing spatial coupling.

The more natural σ choice and a higher cluster number in (c) allow a separation between the left and right saints, and instead of the red noise-cluster we now get a further cluster for the right curtain.

Further refinement (d) leads to additional separation of the left curtain, the background behind Saint Sixtus, his and Madonnas feet as well as her skirt. Unfortunately, we were not able to separate the Madonna from her child.

(a) Original

(b) $n = 5$, $\tau = 20$, $\sigma = 35$
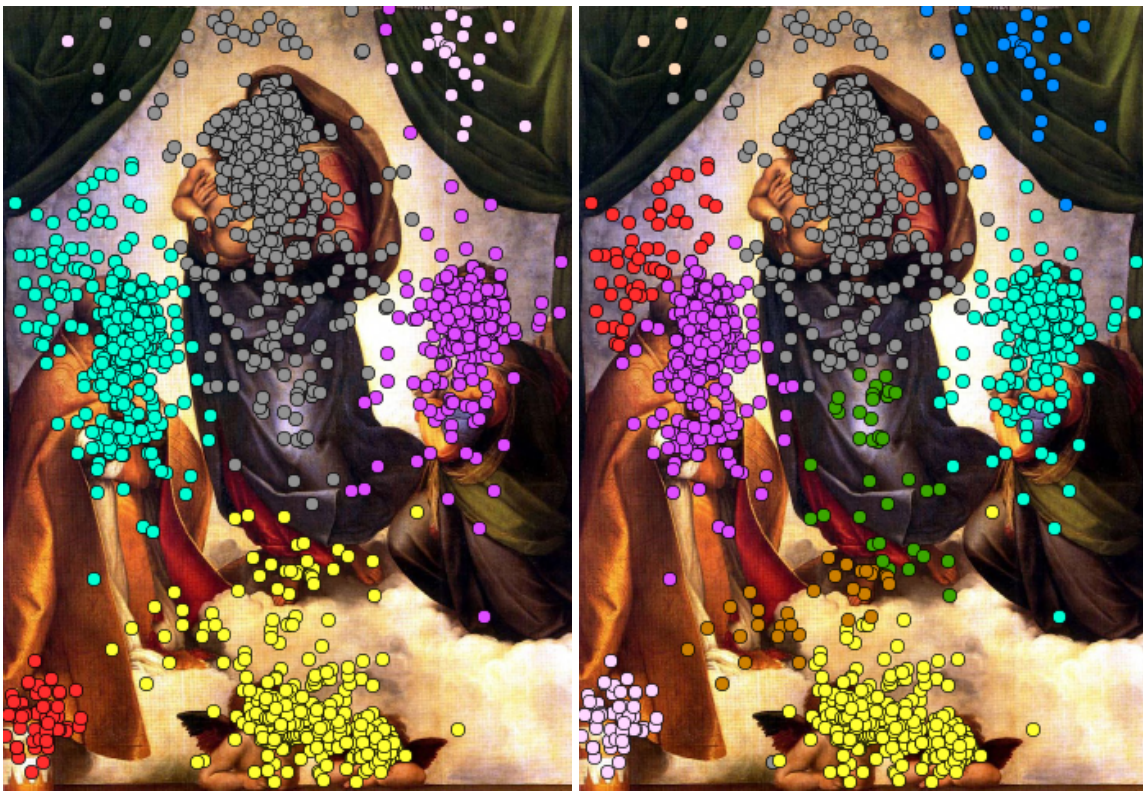
(c) $n = 6$, $\tau = 50$, $\sigma = 60$

(d) $n = 10$, $\tau = 60$, $\sigma = 50$

Figure 4.1: The Sistine Madonna (1512) by Raphael

<table>
<tr><td>(a) Original</td><td>(b) $n = 6$, $\tau = 100$, $\sigma = 250$</td></tr>
<tr><td>(c) $n = 7$, $\tau = 80$, $\sigma = 120$</td><td>(d) $n = 9$, $\tau = 80$, $\sigma = 120$</td></tr>
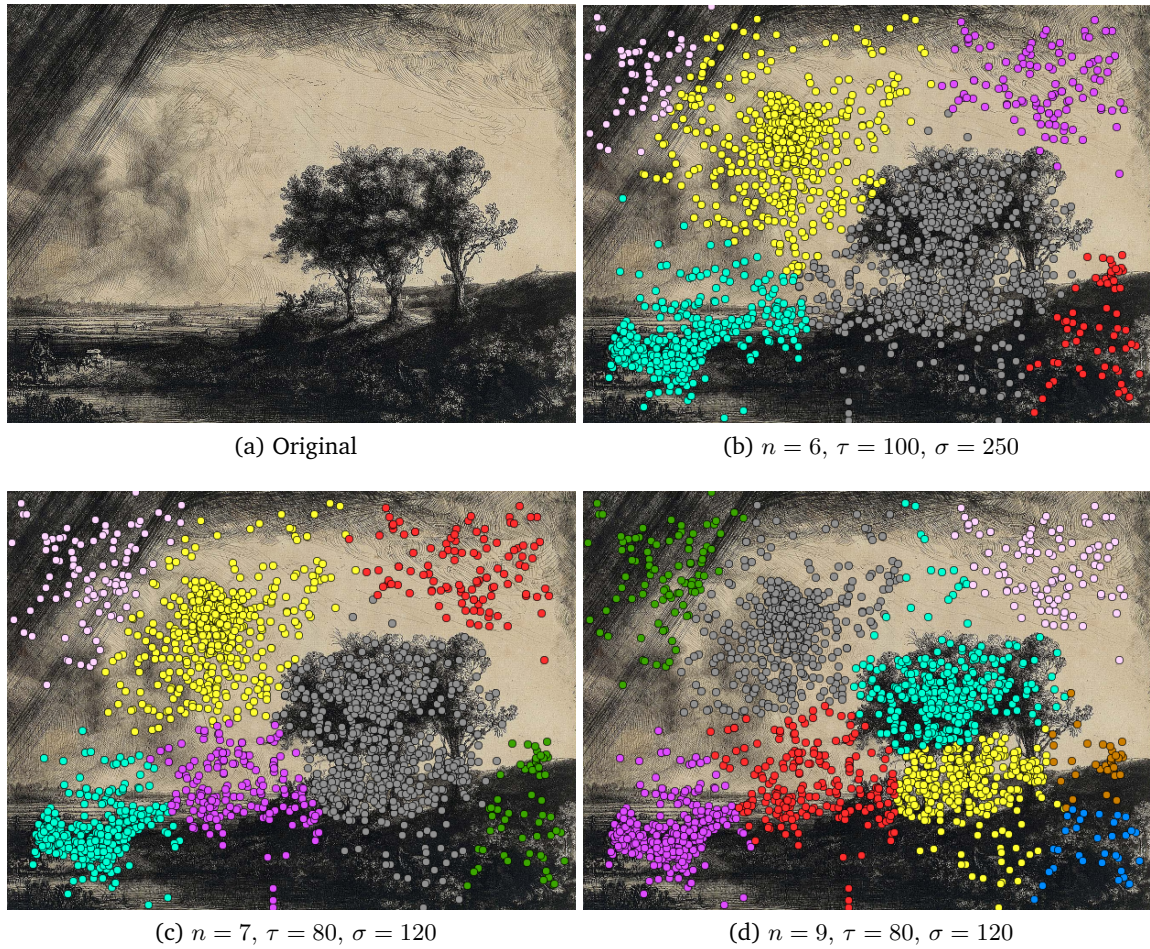</table>

Figure 4.2: The Three Trees (1643) by Rembrandt

Three Trees

In Figure 4.2(a), we see Rembrandt's "Three Trees", widely regarded as his greatest and most elaborate landscape etching. It features a combination of many details: Of course there are the three trees as most prominent feature in the foreground, complemented by a fisherman accompanied by his wife in the lower left, a painter on the hill to the right of the trees and a cloud formation resembling and upcoming storm to their left.

Already the coarse clustering (b) shows a good separation of the trees, also assigning clusters to the fisherman and the stormy clouds, additionally the three corners of the image.

The refinement (c) now also separates the fisherman and his wife from the background, and further refinement (d) even separates the treetops from their trunks, also assigning an own cluster to the painter.
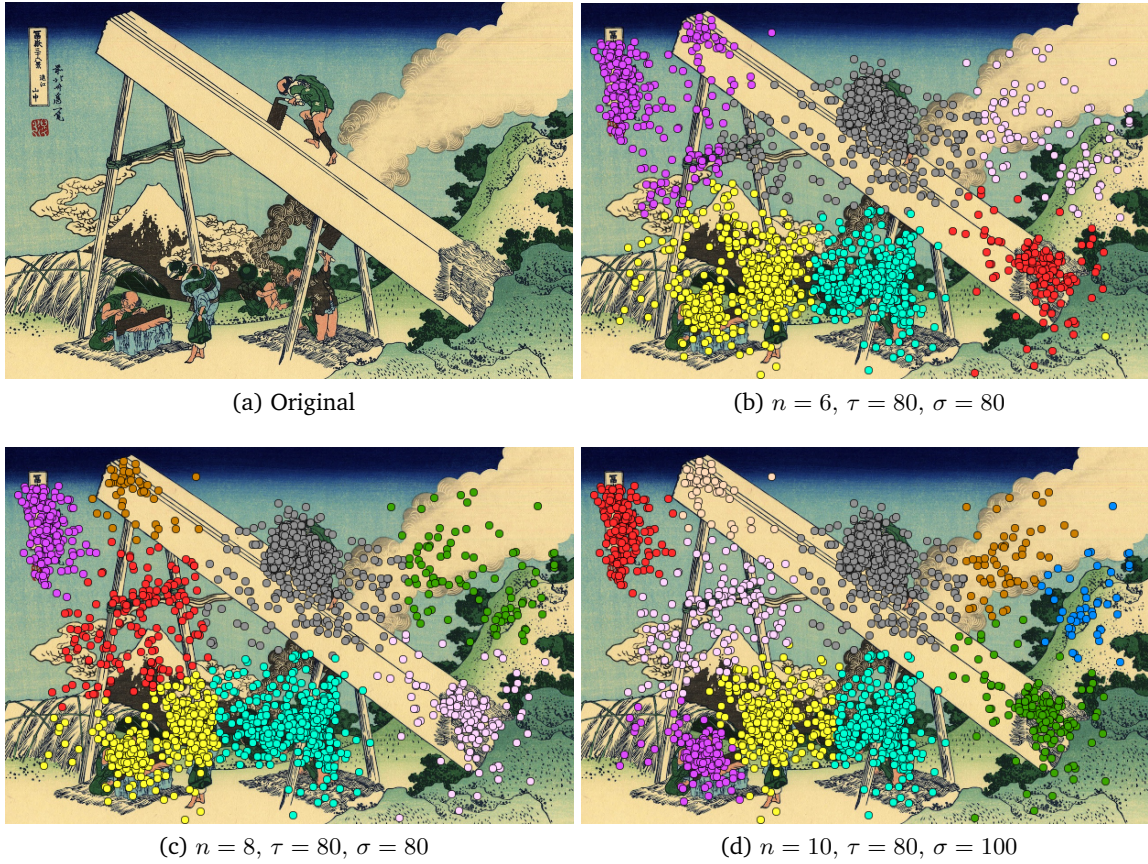
(a) Original

(b) $n = 6$, $\tau = 80$, $\sigma = 80$

(c) $n = 8$, $\tau = 80$, $\sigma = 80$

(d) $n = 10$, $\tau = 80$, $\sigma = 100$

Figure 4.3: In the Totomi Mountains (1830) by Hokusai

In the Totomi Mountains

Figure 4.3(a) shows Katsushika Hokusai's "In the Totomi Mountains", one out 36 color woodblock prints of his series "Thirty-six Views of Mount Fuji", featuring the prominent volcano Mount Fuji, the highest mountain in Japan. This picture was chosen for it has a large number of distinct features: Two men sawing a large tree block on stands, a woman with a baby on her back watching a third man sawing a tree trunk on the ground and another man in the back making a fire, the Mount Fuji in the back entwined by a curling cloud.

The clustering into six clusters (b) focuses on a rough separation of the people (three clusters on five people), the legend in the top left, the smoke of the fire and the strawmat at the foot of the tree block. One might object that none of these clusters separate any of the objects properly, but as we impose just six clusters one can't expect a clear separation as the objects in between have to become assigned as well.

Increasing the cluster number to eight (c), we gain an splitting of the former upper left cluster, now properly isolating the legend, the smoke around the Mount Fuji with its peak and the top of the wood block.

Further refinement to ten clusters (d) splits the two lower clusters into three, now distinguishing between the man sawing on the ground, the woman with her baby and the standing woodcutter with the man making the fire in the background. Additionally, the upper right cluster gets split into two, now correctly differentiating between the smoke and the hill.

5 Discussion

Given a good choice of parameters, the algorithm showed to be able to cluster the points to the subjectively identified objects in the picture, backing up the hypothesis that human sight exhibits the metastable behaviour on recognized objects.

This conclusion already implies the current problem of the approach, the parameters. Although most of the resulting clusters proved to be stable in many situations, high cluster numbers or small σ values lead to unstable results.

The time step parameter $\Delta\tau$ could be completely eliminated by using a time-continuous Markov chain as underlying transition model, leading to a transition rate matrix. Unfortunately the ad hoc approach using

$$W_{ij} = \left(\frac{\sum_{a \to b} M_{ai} M_{bj} \tau_{a \to b}}{\sum_{a \to b} M_{ai}} \right)^{-1},$$

denoting by $\sum_{a \to b}$ the sum over all fixation transitions, from a to b, and $\tau_{a \to b}$ the corresponding transition time, leads to worse results. It is not yet clear to the author how to construct the most likelihood estimator for the corresponding process.

A, compared to the here chosen maximum method (4.2), more sophisticated method to discretize the fuzzy clustering could also proof crucial in improving the results. One possibility might be weighting the clusters with their size, thus, emphasizing their relative shape and/or discarding points with no clear assignment to an extra cluster. This would also solve the problem of non-assignment of some clusters mentioned in section 4.2.

The choice of other *mass matrices M* in (4.1) and a different weighting of the states than by the stationary distribution π in section 3.1 might allow for further adjustment of the results.

Overall, the presented method, turning the time series into a Markov chain and applying PCCA+ for the clustering, proved to be successful and a viable approach to spatial time series clustering with open opportunities for further improvement.

References

[1] P. Deuflhard, W. Huisinga, A. Fischer, and Ch. Schütte. Identification of almost invariant aggregates in reversible nearly uncoupled markov chains. *Linear Algebra and its Applications*, 315:39–59, 2000.

[2] P. Deuflhard and M. Weber. Robust perron cluster analysis in conformation dynamics. *Linear Algebra and its Applications*, 398:161–184, 2005.

[3] W. Huisinga. *Metastability of Markovian systems: A transfer operator based approach in application to molecular dynamics*. PhD thesis, Free University Berlin, 2001.

[4] D. Kressner. Block algorithms for reordering standard and generalized schur forms. *ACM Transactions on Mathematical Software*, 2006.

[5] S. Kube and M. Weber. A coarse graining method for the identifiation of transition rates between molecular conformations. *Journal of Chemical Physics*, 126(2), 2007.

[6] J. A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965.

[7] Jan-Hendrik Prinz, Hao Wu, Marco Sarich, Bettina Keller, Martin Senne, Martin Held, John D. Chodea, Christof Schütte, and Frank Noé. Markov models of molecular kinetics: Generation and validation. *Journal of Chemical Physics*, 134, 2011.

[8] S. Röblitz and M. Weber. Fuzzy spectral clustering by PCCA+: application to markov state models and data classification. *Advances in Data Analysis and Classification*, 7:147–179, 2013.

[9] M. Weber. *Meshless methods in Conformation Dynamics*. PhD thesis, Free University Berlin, 2006.

[10] M. Weber and K. Fackeldey. G-PCCA: Spectral clustering for non-reversible markov chains. *submitted*, 2015.

[11] M. Weber and T. Galliat. Characterization of transition states in conformational dynamics using fuzzy sets. *ZIB-Report*, 02-12, 2002.