Automatically Extracting ERP Component Latencies Using a Template Matching Algorithm

Sven Lesche1

1 Ruprecht-Karls-University Heidelberg

Author note

Author Notes go here.

The authors made the following contributions. Sven Lesche: Conceptualization, Writing - Original Draft Preparation, Writing - Review & Editing.

Correspondence concerning this article should be addressed to Sven Lesche, Im Neuenheimer Feld 695, 69120 Heidelberg. E-mail: [sven.lesche@psychologie.uni-heidelberg.de](mailto:sven.lesche@psychologie.uni-heidelberg.de)

Abstract

One or two sentences providing a **basic introduction** to the field, comprehensible to a scientist in any discipline.

Two to three sentences of **more detailed background**, comprehensible to scientists in related disciplines.

One sentence clearly stating the **general problem** being addressed by this particular study.

One sentence summarizing the main result (with the words “**here we show**” or their equivalent).

Two or three sentences explaining what the **main result** reveals in direct comparison to what was thought to be the case previously, or how the main result adds to previous knowledge.

One or two sentences to put the results into a more **general context**.

Two or three sentences to provide a **broader perspective**, readily comprehensible to a scientist in any discipline.

*Keywords:* keyword, more keyword, crazy keyword

*Word count:* X

Automatically Extracting ERP Component Latencies Using a Template Matching Algorithm

# History of Extraction Methods

## Importance of ERPs

In cognitive neuroscience, the accurate extraction of latencies of event-related potentials (ERPs) stands as a crucial step in understanding the timing of neural processes underlying cognitive functions. In studies focusing on individual differences it is paramount to objectively, reliably and efficiently extract ERP latencies. Algorithms allow efficient and objective approaches to latency extraction, but often prove difficult due to ERPs with low signal-to-noise ratios. To deal with this issue, researchers often forgo algorithms and manually inspect each ERP, identifying patterns reflecting the component of interest by hand. Manual extraction is a labor-intensive and time-consuming endeavor, often fraught with subjectivity. This paper introduces a novel algorithm for the automatic extraction of ERP latencies using template matching. We aim to show that our algorithm improves on existing approaches and enables more efficient, objective, reliable and valid extraction of ERP latencies.

## Prior Algorithms

### Peak Latency.

Automatically extracting component latencies has long been a goal in ERP research. *Peak latency* algorithms present the earliest attempt at automating the latency extraction process[REF]. The peak latency approach involves finding the point in time within a set measurement window that has the largest voltage deflection in the appropriate direction. use of this method is widespread partially due to the ease of its implementation and low computational requirements. **something about historical ease of use** **Luck, 2014, p. 286 cites Donchin & Heffley, 1978 as historical example**. However, ease of implementation is accompanied by several drawbacks. Peak latency algorithms can be corrupted by several factors.

#### Issues with peak latency.

Firstly, peak latency approaches are blind to the general structure within the ERP signal. They locate the point in time with the largest voltage deflection within the measurement window. The “true” maximum signal may lay just outside of the measurement window, resulting in the algorithm picking an outside edge of the window, even though the signal increases in amplitude just following that time-point. [REF; Luck 2014 mentions this I think] have modified the simple *peak latency* algorithm to only consider maxima inside the measurement window which are also larger than the surrounding data-points. A spike just on the edge of the measurement window will not meet these criteria. This approach protects against the influence of noise or surrounding components on the edges of the measurement window, but remains vulnerable to high frequency noise inducing peaks inside the measurement window.

The maximum voltage deflection may not reflect the true point in time at which the process underlying the component reaches its maximum. Rather it may be the result of high frequency noise inducing a spike in the signal, independent of any cognitive process. This is problematic especially in later ERP components, as the broader measurement window most commonly applied increases the likelihood of high frequency noise inducing the maximum voltage deflection ([Clayson et al., 2013](#ref-clayson2013noise)).

The sensitivity of the peak latency approach to the size of the measurement window specified is further increased by the *superimposition problem* ([Luck, 2014](#ref-luck2014introduction)). In larger measurement windows, later components may also already influence the amplitude on the outer ranges of the measurement window. *peak latency* algorithms can thus fail to pick the *local maximum* of a component of interest in favor of a peak possibly influenced by later or earlier components. Researchers have to carefully choose the measurement window in order to include most of the signal related to the component of interest while simultaneously excluding influences of other components.

Lastly, as Luck ([2014](#ref-luck2014introduction)) so aptly states: *There is nothing special about the point at which the voltage reaches a local maximum*. The largest deflection does not inherently relate to any physiological or psychological process and may not even reflect the true maximum of the component of interest. Luck ([2005](#ref-luck2005ten)) visually demonstrates how peak latency may be a result of the overlap of multiple components and not related to any single component.

### Fractional Area Latency.

*Fractional Area Latency* approaches hope to remedy some of the problems associated with *peak latency* algorithms. This technique revolves around the area under the ERP signal in a given measurement window. The goal is to find the point in time that divides that area under the signal into a given fraction to the left and right of it. The time-point halving the area under the signal, for example, is referred to as the *50% area latency*. This approach is much less susceptible to the influence of high frequency noise, as short spikes in the signal do not have a strong impact on the area under the signal ([Liesefeld, 2018](#ref-liesefeld2018estimating)). It remains highly dependent on the measurement window ([Luck, 2014](#ref-luck2014introduction)). Choosing a shorter window may result in only a part of the area of the component of interest being captured. A wider measurement window on the other hand might be influenced by surrounding components. *Fractional Area* measures thus work best for investigating an isolated component ([Luck, 2014](#ref-luck2014introduction)), limiting the applicability of algorithms based on this.

### Jackknifing.

Another approach towards dealing with noisy subject-level ERPs is to try and mitigate that noise by averaging multiple subject-level ERPs. This technique is referred to as *jackknifing* ([Miller et al., 1998](#ref-miller1998jackknife)). It uses the grand average of subject-level ERPs and generates sub-grand averages by removing one of the subject-level ERPs. This averaging procedure results in ERPs with higher signal-to-noise ratios. Both *peak* and *area* based measures can then be applied to the jackknifed data to extract latencies. As any two jackknifed signals share percent of the subject-level signals that are averaged with each other, each jackknifed sub-average is quite similar to all others. This artificially decreases the error variance, which needs to be corrected for when testing for differences between groups ([Ulrich & Miller, 2001](#ref-ulrich2001using)). Because latencies extracted from jackknifed ERPs are based on averaged waveforms, they can not readily be associated with any single subject, preventing this method from generating individual-level latency estimates needed for individual differences research. This problem was addressed by ([Smulders, 2010](#ref-smulders2010simplifying)) who introduced a transformation able to generate individual-level latency estimates.

## Comparison of Algorithms

Kiesel et al. ([2008](#ref-kiesel2008measurement)) applied these different approaches, as well as a few additional approaches not mentioned here, to a variety of ERP components. They simulated latency differences of the visual and auditory N1, the N2pc and the P3 and frequency-related P3 and tested single-participant approaches and jackknife-approaches combined with peak latency, fractional area latency, relative criteria and baseline deviation methods on their ability to detect these effects. The most widely used technique of single-participant approaches combined with peak latency extraction proved to not be the most efficient method to detect latency effects. This becomes especially true as the signal-to-noise ratio decreases. Overall, jackknifing ERPs and using the relative criterion technique or the fractional area latency technique was shown to be the best approach across the components and datasets they analyzed.

Kiesel et al. ([2008](#ref-kiesel2008measurement)) were concerned with extracting condition differences in ERP research. While this is important, techniques applied in individual differences research have to extract latencies from subject-level ERPs and produce both reliable and valid estimates. They did not assess psychometric properties of latency estimates. Addressing this gap in research, Sadus et al. ([2023](#ref-sadus2023multiverse)) investigated the influence of different preprocessing strategies and latency extraction techniques on psychometric properties of the latency values generated and the ability to detect an age-related effect in P3 latency. They varied the strength of the low-pass filter applied to the data, used both single-participant and jackknife approaches and extracted latencies either automatically or manually, using either a peak latency or area latency approach. Both the size of the effect and the psychometric properties, such as reliability or homogeneity of the latency values, varied between the different analysis strategies. No combination of preprocessing steps and extraction method proved best across all tasks and groups. A total of 7 out of 40 possible pipelines generated consistently desirable reliabilities (), homogeneities () and effect sizes (). All of those seven pipelines used manual extraction methods either based on peak or area latencies. While automated extraction methods would improve both efficiency and objectivity, fully automated approaches failed to generate consistently reliable and valid latency measures ([Sadus et al., 2023](#ref-sadus2023multiverse); [Schubert et al., 2023](#ref-schubert2023robust)). Yet, manual extraction methods are highly time-consuming and impede reproducibility. We hope to show that our algorithm can match the performance of manual extraction while providing a more efficient and objective approach for extracting individual component latency values.

# Our algorithm

## Motivation

The shortcomings of prior algorithms may be due to none of them aiming to replicate human behavior and rather focusing on generating similar decisions, not similar decision processes. Our proposed algorithm more closely resembles the process expert ERP researches employ.

## template matching

### Introduction.

During latency extraction, most researchers have some mental representation of what the component *should* look like and where it *should* generally appear. When visually inspecting ERP signals, their goal is to identify a pattern within the signal that resembles the component of interest in shape, size and location.

Finding a given pattern inside a noisy signal is not a novel task. Algorithms aiming to detect the appearance of a pattern - a *template* -inside audio-, video- or radio signals have been around for over 50 years. And a large amount of research has gone into optimizing these *template matching* algorithms ([Briechle & Hanebeck, 2001](#ref-briechle2001template); [Brunelli, 2009](#ref-brunelli2009template); [Mahalakshmi et al., 2012](#ref-mahalakshmi2012image); [**lewis1995fast?**](#ref-lewis1995fast)).

No matter the implementation details of a particular template matching algorithm, all aim to answer the question “Does this (smaller) template appear in my (larger) signal?”. To achieve this, a researcher needs to specify two things. First, the template they want to search the signal for. And second, the *similarity measure* which quantifies how well the template fits in a given spot of the signal.

### Similarity measures.

Specifying the template is mostly a philosophical question dependent on the specific task and type of signal. Choosing a similarity measure on the other hand is much more methodological. Across a number of papers, several different similarity measures have been proposed. They follow one of two general lines of thought ([Brunelli & Poggiot, 1997](#ref-brunelli1997template); [Goshtasby et al., 1984](#ref-goshtasby1984two)). The first type of similarity measure aims to minimizes some value reflecting the distance between template and the signal. The second type aims to maximize some form of correlation between signal and template. We have chosen to implement algorithms based on two of the possible measures, one minimizing the Sum of Squared Differences (MINSQ) and one maximizing the correlation (CORR) between the template and the signal. We wanted to implement both a similarity measure following a traditional distance-minimization approach and a correlational approach in order to gauge the efficacy of these approaches when applied to our field of research.

### Template generation.

Finding an appropriate template is more challenging. Depending on the field, the template to search for is easily specified. If you are looking to extract a particular audio-signal from a recording or some specific object in an image you can easily use that object as a template. The difficulty increases if it is not exactly certain what template you are looking for. Recent research in image processing has attempted to use template matching to process faces ([Brunelli, 2009](#ref-brunelli2009template)). You cannot just use “the ideal set of eyes” to identify a face. Each person comes with their own set of eyes, different from all others in some quantifiable way. A similar issue accompanies attempts of template matching approaches in ERP research. The variance in ERP signals introduced by the task or the sample of participants hinders a successful implementation of template matching algorithms using only one idealized template over all types of studies. Finding a template that reflects the influence of the task or the sample, equivalent to finding the person who’s eyes you are looking for, will significantly improve performance.

## Prior attempts

There have been some attempts at using an idealized signal structure as a template to identify ocular artifacts in noisy subject-level data ([Li et al., 2006](#ref-li2006automatic)) or to predict subject behavior on a single trial level ([William et al., 2020](#ref-william2020erp)). However, these approaches were not concerned with estimating the timing of components, but only interested in detection of a specific signal.

Borst and Anderson ([2015](#ref-borst2015discovery)) and Anderson et al. ([2016](#ref-anderson2016discovery)) developed a machine-learning approach that aims to discover cognitive processing stages on a single trial level. In a first step, their algorithm makes use of multivariate pattern analysis to detect “bumps” in the EEG signal representing the onset of a new cognitive state. They assumed that entry into a new state would be accompanied by a spike in all electrodes similar to a 50 ms half-sine. This 50 ms half-sine then serves as a template with which their algorithm tries to detect those “bumps” in activity. However, their assumptions regarding the template and the location of activity are somewhat crude generalizations made necessary by noisy single trial data. Using template matching to extract component latencies from ERPs requires a more informative template.

## our solution

A simple approach towards designing a more informative template would be to generate an idealized component structure. Prior knowledge about the shape, size and location about the component of interest could then be made use of. One could draw up “the perfect P3” and attempt to use this as a template. Nonetheless, this would neglect the experiment-specific and task-specific variance in the morphology of ERP components, resulting in a template that does not optimally reflect the data. We addressed this problem by using the grand average as an experiment-specific template of the component of interest that will reflect influences of the task and sample on the morphology of the idealized ERP component. It has a higher signal-to-noise ratio than any individual subject-level ERP while still being influenced by experiment-specific changes in component morphology. Researchers can (and already often do) use the grand average to gather insight into the time window in which the component of interest occurs ([Kiesel et al., 2008](#ref-kiesel2008measurement)). The grand average and the specific window in which the component of interest occurs is used as a template and matched to subject-level ERPs.

### Differences to other template matching studies.

The grand average is the mean of all subject-level ERPs and thus minimizes the sum of squared deviations between each subject-level ERP and itself. Thus, across all subjects, it is the best approximation for each subject-level ERP. However, individual differences in the shape, size and location of the component of interest remain. The goal of our algorithm is to quantify these individual differences. This is similar to facial recognition not only detecting the presence of eyes, but also determining the person the eyes belong to. We do not only aim to detect the presence of a component, but also aim to measure individual differences in shape, size and location of the component.

To achieve this, we introduce variability into the template reflecting individual differences in the morphology of the component. Determining which transformation of the template best fits a given signal will allow us to investigate individual differences.

## Subject-level transformations as the goal

We introduce variability in the template through two sources, amplitude and latency. Variability in amplitude is controlled by the parameter , variability in latency by the parameter . Amplitude of the template is varied by multiplying the whole template-signal by parameter . Latency is varied by “stretching” or “squishing” the template along the x-axis [See Figure?]. The parameter controls the strength of this transformation. Possible transformations of the template are then compared to the signal and their similarity is evaluated. Recovering the transformation parameters that lead to the best match between template and signal thus allows us to describe individual differences in the latency of a component through the transformation parameter for the participant . In this manner, component latencies can be recovered. Researchers need to specify a time-point of the grand average denoting the latency of the component of interest. The optimal transformation parameters are then applied to the grand average latency to recover the subject-level latency of the component.

## Why our algorithm may perform better

This algorithm aims to address some of the issues faced by other algorithms. It makes use of the entire component structure that is used as a template. This reflects the decision process of expert ERP researchers and enables an intuitive understanding of the decisions made by our algorithm. Because the similarity measures take the whole component structure into account, they are robust to peaks in the signal introduced by high frequency noise. Furthermore, the measurement window set by the researcher only impacts the size and shape of the template. It has no direct connection to the subject-level ERP. The influence of measurement windows on the latencies extracted should thus be lower than in peak or area latency algorithms. Lastly, it is important to note that this is not a machine learning algorithm with a neural net representing some “black box” decision making algorithm. Simplicity and traceability of the decision process was an express goal, allowing more insight into the benefits and drawbacks of our algorithm.

## The present study

In order to compare the quality of our proposed algorithm we will reanalyze the same data analyzed by ([Sadus et al., 2023](#ref-sadus2023multiverse)). We can compare the psychometric properties of our algorithm to those of previously established algorithms, investigate the impact of different preprocessing steps and evaluate the correlation between latencies extract by our algorithm and those extracted manually be an expert ERP researcher.

In their study, Sadus et al. ([2023](#ref-sadus2023multiverse)) extracted latencies of the P3b component, henceforth simply referred to as P3. The P3, is a centroparietal positive-going component, peaking around 300 ms after stimulus onset. It is a late higher-order cognitive component [REFS] associated with [Processes]. A number of studies have demonstrated a large effect of age on the latency of the P3 across a number of tasks [REFs]. In a multiverse approach Sadus et al. ([2023](#ref-sadus2023multiverse)) tested several extraction methods with varying preprocessing steps in their ability to detect this age effect. They used three tasks, each measuring one of the executive functions proposed by Miyake et al. ([2000](#ref-miyake2000unity)). To measure the functions *updating*, *shifting* and *inhibition*, they employed an Nback, a Switching and a Flanker Task, respectively. Studying three different tasks allows us to gain insight into a larger variety of higher-order cognitive processing, improving the generalizability of our findings.

Nonetheless, we will restrict our analysis to extracting P3 latencies. It usually has a broad and isolated structure with comparatively low influence of surrounding components [REF]. This makes the P3 one of the easier components for automated latency extraction approaches. After we can demonstrate proof-of-concept for P3 latency extraction, we will evaluate the algorithms ability to be extended to other ERP components.

We hope to show that a template matching approach using the grand average as a variable template can successfully extract P3 component latencies. Ideally, use of this algorithm will improve psychometric properties in comparison to prior algorithms, show high correlations with manually extracted data and present an objective and efficient way to extract ERP latencies.

# Algorithm

## Implementation in Matlab with details

We implemented the algorithm in MATLAB [VERSION, CITATION]. Data must be in ERPLAB - shape **better way**. The researcher then specifies the name and polarity of the component of interest as well as the measurement window. This window is used to extract the template. In order to transform the template, we use MATLABs Curve Fitting Toolbox [Citation] to generate *sum of sines* functions that fit to the data with . We then add an amplitude parameter to the overall function in order to allow for scaling of the amplitude of the template and a frequency parameter to all frequency terms in the *sum of sines* function to allow for “squishing” or “stretching” the template along the x-Axis. The template is described by a function with sine terms and their respective amplitude , frequency and phase .

As these transformations also change the measurement window, we chose to use the subject-level ERP as a template and keep the grand average untransformed as a signal. This reverse matching approach is only an implementation detail and does not affect any decisions made by the algorithm.

Depending on the similarity measure employed, we use different functions to find the set of optimal parameters that lead to the transformation of ERP of subject most similar to the grand average.

### MINSQ.

The MINSQ algorithm minimizes the weighted sum of squared differences between the transformed subject signal and the grand average.

This weighting vector is computed as follows

Where

denotes the measurement window, the signal strength of the th element, the maximum voltage deflection inside the measurement window and the time of the th element.

This places more emphasis on fitting the template within the measurement window specified and to places in the signal where the voltage deflection is high.

We use MATLABs *fit* function to find optimal parameters with upper and lower bounds such that . As this function may be prone to converging on local minima, we initialize 5 different start points. The solution with the best correlation between transformed template and signal that multiple start points converged on is selected.

In cases where the subject-level ERP only has signal with deflections opposite of the deflection of the component of interest, it may occur that the parameter . In these cases, we attempt to re-match the signal with an added parameter shifting the entire template up or down.

Should the algorithm again converge on a solution with , the latency value is set to NA.

### CORR.

The CORR algorithm optimizes the parameters to produce the maximal correlation between the transformed subject-level signal and the grand average for values in the measurement window.

represents the vector of values of the transformed subject ERP that are in the measurement window, the vector of values of the grand average that are in the measurement window.

MATLABs *fminbnd* function finds the optimal transformation parameter maximizing . This function will estimate the objective function for all values inside the given bounds and converge on the global optimum. We do not need to initialize a number of different starting points here.

Both algorithms then return the transformation parameters that result in optimal similarity of transformed template and signal. We use the returned value of the parameter to transform the component latency specified by the researcher in the grand average to the component latency of the subject-level ERP signal .

## Review methods

Researchers can manually review all choices the algorithm has made in a custom-built user interface. For both algorithms, we used the correlation between transformed template and signal as a fit-index. This can be used to only review those cases where the correlation between template and signal dips below a certain value, indicating low similarity between matched template and signal. We will investigate the additional benefits a manual review process provides over accepting the choices as-is or only automatically discarding those matches with correlations .

# Data

The data used here were first published by [Sadus 2023] and are a subset of the data collected by [Löffler REF].

## Participants

The data is comprised of 30 young participants (18-21 years old, mean age = 19.37, SD age = 0.76) and 30 old participants (50-60 years old, mean age = 55.83, SD age = 2.87). This sample was part of a larger study [Löffler REF] of which the 30 youngest and 30 oldest participants were selected. All participants had normal or corrected to normal vision. None of the participants had neurological or mental disorders, used psychotropic drugs, wore a pacemaker or suffered from red-green color vision deficiency. The participants provided informed consent prior to participation and received 75€ or course credit for participation. **ethics stuff?**

## Tasks

Participants completed a set of 3 tasks: a Flanker Task, an Nback Task and a Switching Task. These tasks each measure one of the executive functions proposed by [Miyake, 2000]. [Löffler] programmed all tasks in MATLAB [ref] using the software package Psychtoolbox [version 3-0.13, REF]. Stimuli were presented centrally on a black background. We instructed participants to respond as quickly and accurately as possible.

### Flanker Task.

We administered a standard Arrow Flanker task [Eriksen & Eriksen 1974] to measure participants’ *inhibition* ability. A central arrow pointing either to the left or to the right is flanked by two additional arrows to each side. These flanking arrows either point in the same or in the opposite direction as the central arrow. Participants have to respond by button press in which direction the central arrow pointed, disregarding the congruent or incongruent flanking arrows. Participants completed a set of practice trials and a total of 100 congruent and 100 incongruent trials.

### Nback Task.

We administered an adapted version of the Nback task from [Scharinger 2015] to measure participants’ *updating* abilities. A stream of letters is shown to the participant. In the 0-back condition, participants have to indicate by keypress whether the presented letter is equivalent to a target letter. In the 1-back condition, participants have to indicate whether the currently presented letter is the same as the letter presented one trial before or not. [Löffler] also had participants complete a 2-back condition. We excluded this condition from our analysis as it did not produce clear ERPs. Participants completed a set of practice trials and a total of 96 trials per condition.

### Switching Task.

We administered a Switching task to measure participants’ *shifting* ability. A stream of colored digits ranging from 1 to 9 was presented. Participants had to indicate whether the digit was greater than or less than 5 or whether the digit was odd or even depending on the color of the stimulus. A colored fixation cross just prior to stimulus presentation cued the rule participants had to follow in the upcoming trial. Participants had to either follow the same rule as in the trial before or switch to the other rule. Participants completed a set of practice trials and 192 trials each in the repeat and in the switch condition.

## Procedure

The original study [Löffler] consisted of three test sessions. The three tasks this study focuses on were all administered in the first session. The second session also included EEG measurement with 3 additional tasks. The third session was used to measure intelligence and working memory capacity. No EEG measurements were taken here. In sessions including EEG measurements, participants were seated approximately 140cm away from a monitor in a sound-attenuated room.

## EEG recording and processing

EEG was recorded using 32 Ag/AgCl scalp electrodes placed in the 10-20 system. Additional electrooculogram (EOG) measures were taken by two electrode placed above and below the left eye to correct for ocular artifacts. All impedances were kept below 5 kΩ. The signal was recorded with a sampling rate of 1000 Hz (band-pass 0.1 Hz - 100 Hz) and online-referenced to Cz. Following data acquisition, the raw data was down-sampled to 250 Hz. To remove artifacts we conducted an ICA on a cloned version of the dataset down-sampled to 100 Hz and passed through an additional high-pass filter of 1 Hz. Both the original down-sampled data as well as the ICA-dataset were cleaned by removing channels with unusually long flatlines, artifact-rates or line-noise. Channels removed were interpolated following this procedure and the data was re-referenced to the average across electrodes. ICA was conducted using the InfoMax algorithm and the resulting decomposition applied to the original dataset. ICs were labelled using the ICLabel Algorithm [Pio-Tonachini et al. 2019, aus sadus] and removed if the IC was less than 50% likely to be brain activity. We then applied Butterworth low-pass filters with varying cutoff frequencies (8 Hz, 16 Hz, 32 Hz) and a roll-off of 12 dB/octave. Data was segmented into 1200ms long segments starting 200ms before stimulus onset. Segments containing artifacts were automatically detected and removed. As a last step, we conducted a baseline correction using the 200ms prior to stimulus onset.

## ERP analysis

Analyses were conducted in MATLAB [Version, citation]. We only included correct trials into analysis. We investigated the ERPs containing the P3b at the electrode Pz, similar to previous literature [REFS].

### Latency extraction.

To evaluate the impact of the specified measurement window, we extracted latencies three separate times using either a narrow (250-600ms), medium (200-700ms) or wide (150-900ms) measurement window. We used the peak latency approach to determine the latency of the P3 in the grand averages. We applied our algorithms using both the distance-based (MINSQ) and correlation-based (CORR) similarity measures to the data and obtained transformation parameters and fit values. Subject-level latencies were recovered from the respective grand average latency and the subject-specific transformation parameters. We also ran a traditional simple peak latency algorithm and a fractional area latency algorithm determining the 50% area latency. This was either done in an uninformed fashion, using the measurement window specified beforehand or with an adapted measurement window, based on the transformation parameters returned by the template matching algorithms. If the CORR algorithm returns the parameter for participant , for example, the medium measurement window of this participant would be transformed from 200-700ms to 245-745ms. To investigate the benefits of manually reviewing the decisions of the algorithm, we chose to review all matches that resulted in fit values . We then inspected the subject-level ERP with the matched template displayed over it and either accepted, rejected or manually determined the P3 latency of these ERPs. We also explored the impact of automatically excluding all those matches with fit values .

In our dataset, each of the 60 participants contributed 6 ERPs per task to the data. One ERP averaged over all trials of each of the two conditions and two more ERPs generated from an odd-even split on a trial level of that condition. These 360 ERPs each from the 3 different tasks were passed through 3 different low-pass filters and subjected to analyses with 3 separate measurement windows. We applied both the correlation-based (CORR) and distance-based (MINSQ) algorithm and either reviewed the results manually, discarded bad matches automatically or accepted the results regardless of fit. We also applied both a peak latency and area latency algorithm with either uninformed measurement windows or measurement windows transformed by the transformation parameters recovered from the CORR or MINSQ algorithm. This results in 3 tasks 3 filters 3 windows 12 algorithms = 324 different extraction pipelines.

# Validation Techniques

We investigated the impact of latency extraction method on several psychometric properties as well as the effect size of the age effect. We also compared the methods in their correlation with manually extracted latencies, which can be seen as a benchmark for proper latency extraction [Sadus].

## Reliability

We estimated reliability by computing Spearman-Brown corrected split-half correlations of ERPs generated from an odd-even split at the trial level.

## Homogeneity

To compute a methods homogeneity , we calculated its correlation with all other methods and took the mean of the Fisher-Z transformed correlation coefficients. Correlation coefficients of 1 cannot be transformed. Thus, we set all correlations to . This mean correlation with other methods indicates the extent to which a particular method reflects the total of all other measures.

## Effect size?

To investigate the effect of age on P3 latencies, we ran a repeated measures ANOVA with the between factor age (young vs. old) and the within factor task (Flanker, Nback, Switching).

## Correlation with manual extraction

To quantify the extent to which an extraction method is able to replicate the benchmark of manual extraction, we correlated the latencies extracted by a particular method with those extracted by an expert researcher in the same task and filter condition

## Impact of Filters / Measurement Window

We also quantified the impact preprocessing steps and the choice of measurement window have on the reliability and validity of extraction methods. We ran a repeated measures ANOVA with the dependent variable being either the estimated reliability of a method or its correlation with manually extracted data and the between factor filter (8 Hz vs. 16 Hz vs. 32 Hz) and the within factor measurement window (narrow vs. medium vs. wide).

# Results

All data wrangling and statistical analyses were using .

## Review process

We reviewed results of the CORR and MINSQ algorithms if their fit was below or if . For the CORR algorithm, out of 9720 ERPs evaluated by the algorithm, we inspected 1045 (10.75 %) of ERPs. Of those ERPs, we rejected 23.35 % and accepted 64.21 % of the results despite their fit. We manually corrected the decisions in 12.44 % of cases. Automatically rejecting fits discards 2.09 % of latencies. For the MINSQ algorithm, out of 9720 ERPs evaluated by the algorithm, we inspected 1063 (10.94 %) of ERPs. Of those ERPs, we rejected 28.22 % of ERPs and accepted 62.65 % of the results despite their fit. We manually corrected the decisions in 9.13 % of cases. Automatically rejecting fits discards 1.43 % of latencies. Because the MINSQ algorithm may fail to find a valid solution if an amplitude parameter of fits the signal best, we discarded 7.13 % of cases. This did not occur in the CORR algorithm.

When reporting the psychometric properties of the algorithms, we will focus on those values passed through manual inspection. Values that were gained from a pipeline ending with the automatic rejection filter are reported in parenthesis. Properties of uninspected pipelines can be found in the respective tables in the appendix.

## Reliability

An overview of Spearman-Brown corrected split-half correlations split by task, measurement window and filter setting can be found in [TABLE]. Across tasks, measurement windows and filter settings the CORR algorithm had mean split-half correlations of .89 for manually reviewed latencies ( .87 for automatically reviewed latencies), the MINSQ algorithm .90 ( .86). Uninformed area latency measures presented even better mean split-correlations .92. The average split-half correlation for values extracted by an expert ERP researcher was for area latency measures and for peak latency measures [Sadus 2023].

## Homogeneity

An overview of a method’s mean correlation with other methods split by task, measurement window and filter setting can be found in [TABLE]. Across tasks, measurement windows and filter settings the CORR algorithm had a homogeneity of .81 ( .80), the MINSQ algorithm .87 ( .86). The homogeneity of uninformed area latency measures was .78.

## Effect size

An overview of the effect size of the age effect detect by a particular method split by task, measurement window and filter setting can be found in [TABLE]. Across tasks, measurement windows and filter settings the CORR algorithm had a mean effect size of .18 ( .18), the MINSQ algorithm .23 ( .23). Uninformed area latency measures presented average effect sizes of .15. The average effect size for values extracted by an expert ERP researcher [Sadus 2023] was for area latency measures and for peak latency measures.

## Correlation with manual rater

An overview of the correlation with latency values extracted by an expert ERP researcher [Sadus 2023] split by task, measurement window and filter setting can be found in [TABLE]. Across tasks, measurement windows and filter settings the CORR algorithm had mean correlations of .89 ( .88) with manually extracted latencies, the MINSQ algorithm .91 ( .91). Uninformed area latency measures had a mean correlation of .85.

## Influence of researcher degrees of freedom

The repeated measures ANOVA with split-half correlation of a particular method as a dependent variable, the between factor filter (8 Hz vs. 16 Hz vs. 32 Hz) and the within factor measurement window (narrow vs. medium vs. wide) showed no effect of filter, .00, for the CORR algorithm, no effect, .00, for the MINSQ algorithm and no effect, .00, for uninformed area latency measures. The measurement windows had an effect of .17 for the CORR algorithm, .00 for the MINSQ algorithm and .31 for uninformed area latency measures. Consult [TABLE] for a full overview.

When the correlation with manually extracted latencies was used as a dependent variable, the effect of filter settings for the CORR algorithm were .00, for the MINSQ algorithm .00 and .00 for uninformed area latency measures. The measurement windows had an effect of .30 for the CORR algorithm, .27 for the MINSQ algorithm and .70 for uninformed area latency measures. Consult [TABLE] for a full overview.

# Discussion

## What we found

Our newly proposed pattern matching algorithms displayed consistently good psychometric properties and showed an improved ability to replicate human extraction behavior over previously established approaches like peak latency or area latency algorithms. Manual extraction has so far proven superior to algorithmic approaches ([Sadus et al., 2023](#ref-sadus2023multiverse)) but presents a time- and resource-intensive process. MINSQ-based algorithms were able to replicate this benchmark almost perfectly while presenting a more objective and efficient approach to latency extraction. The algorithm based on minimizing the weighted squared distance between transformed template and signal (MINSQ) correlated to [RESULT] with manually extracted ERP latencies across tasks and preprocessing steps. Our algorithms were also more robust to the impact of different low-pass filters and measurement windows. Application of our algorithm would increase both replicability and scalability as well as significantly reduce the time and resources researchers need to spend on latency extraction.

### Reliability.

A key comparison to evaluate the effectiveness of our algorithm was to test if it is better than already established, simpler algorithms. In our data, especially area latency measures showed to have consistently acceptable reliability and validity while peak area measures almost always proved worse than area latency measures.

Regarding the reliability of extracted latencies across tasks and preprocessing steps, our algorithms did not prove superior to the area latency approach ([RESULT]). Both the MINSQ ([RESULT]) and CORR ([RESULT]) approaches had lower Spearman-Brown corrected split-half correlations. However, the differences in reliability are quite small and only carry low practical differences. This becomes even less relevant if the researcher uses latent-variable approaches to ERP latency as measurement error is controlled for by the common latent variable [REF].

### Homogeneity.

Latency values extracted by the MINSQ algorithm proved to have the highest average correlation with all other extraction methods [RESULT] across tasks and preprocessing steps. A homogeneity of [RESULT] would be considered [excellent?] [REF]. This indicates that this approach best reflects the total of all other measures. The CORR algorithm also proved superior to previously established extraction methods.

### Validity.

Sadus et al. ([2023](#ref-sadus2023multiverse)) showed that manually extracting latency values is the best approach to ensure good psychometric properties and high power. The ability of an algorithm to extract latency values correlating highly with those extracted by an expert ERP-researcher was therefore of high importance to us. Again, our algorithms proved to have a superior ability to replicate human behavior. The MINSQ algorithm had a mean correlation of [RESULT] with manually extracted latencies across tasks and preprocessing steps. Considering the reliabilites of the two algorithms, [RESULT] and [RESULT], this correlation exceeds the theoretically maximal correlation between their true-score values [REF]. This indicates that some amount of their measurement error covaries. We suspect that this is due to the high conceptual similarity between template matching and manual extraction. Errors in the generation of the template or related to the matching may be similar for human researchers and the algorithm.

The CORR algorithm also outperformed previously established approaches in the ability to replicate human behavior, correlating very highly with manually extracted latencies ([RESULT]). Area latency measures also correlate highly with manually extracted data ([RESULT], [RESULT] for peak latency measures) but failed to match the performance of our new algorithms.

### Robustness of method.

Ideally, the quality of an algorithm extracting ERP latencies is largely independent of choices made by the researcher during preprocessing. To evaluate this, we conducted ANOVAs with either the split-half correlation as an estimate for reliability or the correlation of a particular method with manually extracted latencies as an estimate of validity as the dependent variable. The filter setting (8 Hz vs. 16 Hz vs. 32 Hz) and the measurement window (narrow vs. medium vs. wide) were entered as independent variables. The filter setting used did not have an impact on any of the two measures for any algorithm. The measurement window did have an impact on the reliability of the CORR algorithm [EFFECT], area latency [EFFECT] and peak latency [EFFECT]. It also impacted the correlation with manually extracted latency for the CORR algorithm [EFFECT], MINSQ algorithm [EFFECT], area latency [EFFECT] and peak latency [EFFECT]. Our new algorithms proved to be more robust to influences of the measurement window. This is likely due to the lower conceptual influence of the measurement window in our algorithms. While the measurement window directly impacts the values of each subject-level ERP entered into analysis in prior algorithms, it only determines the template used in our algorithms.

### Objectivity.

Any algorithmic approach to ERP latency extraction will be more objective than manually extracting ERP latencies. So we cannot crown any particular algorithm as more or less objective. The completely autonomous versions of peak latency, area latency or our algorithm with automatic rejection of bad fits are all equally objective. One strength of our approach is the ability for the researcher to inspect a subset of the ERPs based on the fit statistic of the matching procedure. This does introduce some subjectivity into the data.

### Efficiency.

However, this ability of our algorithms to generate a fit statistic indicating the degree of certainty with which the match was made, is a great strength of our new algorithm. Depending on the size of their data and the degree to which researchers want to manually inspect their data, one may choose any cutoff value for the fit statistic and inspect none, a subset or all of the ERPs and the choices made by the algorithm by hand. This feature is not present in any of the previous algorithms.

## Differences MINSQ - CORR

We chose to test out two different approaches to quantifying the degree of similarity between template and signal. One minimizing the weighted squared difference and one maximizing the correlation between the two. Both showed improvements over previous algorithms and we can recommend that both approaches be studied further. We did observe some differences between the two approaches in both procedural factors as well as outcome measures.

### Procedurally.

Procedurally, the largest difference between the two approaches is the optimization algorithm underlying them. Due to the invariance of the correlation of two vectors to scaling in amplitude of one vector, we can reduce the number of free parameters optimized during the CORR approach to one. This allows us to use a more exhaustive optimization algorithm that will find the global optimum in some bounded parameter space without the possibility of converging on some local optimum. This is not the case for the multivariate optimization function needed for the MINSQ approach. Here, we initialize the optimization process at several different starting points and check for convergence on a common solution indicating that this solution represents the true global optimum. This is not ideal and could be improved in the future by implementing a more suitable optimization algorithm or even improving on the one currently used.

The MINSQ algorithm may also converge on solutions where if the subject level signal is largely of a polarity opposite to that of the component of interest [See Figure?]. Although we did extend the variability of the template by a parameter vertically shifting the template to account for these cases, sometimes even the extended version will converge on solutions with non-sensible parameter values. This leads to missing values and unreliable fit statistics in the respective cases. In our data, this happened for [RESULT, (about 7%)] of ERP signals. A large portion of these cases may be considered *unindentifiable* even by an expert researcher due to particularly low signal-to-noise ratios. However, some cases where the component can be identified by a human researcher or the CORR algorithm may be classified as missing by the MINSQ algorithm. We will try to implement some additional measures aiming to reduce the number of cases where the MINSQ algorithm fails to converge on a valid solution in future work.

This leads to the difference in the number of cases classified as missing by the MINSQ and CORR algorithms. While [RESULT] of cases were set to NA after manual inspection of the CORR algorithm ([RESULT] after automatic inspection), [RESULT] of cases were set to NA in the MINSQ algorithm following inspection ([RESULT] after automatic inspection). This tradeoff between better properties of the MINSQ algorithm accompanied by more missing values must be taken into account when selecting which algorithm to use. Depending on the number of participants available and the means of analysis missing values may be detrimental, leading to the CORR algorithm being the preferable choice.

The weighting vector used in the MINSQ algorithm represents another difference between the two algorithms. We used it to reflect the increased emphasis a human researcher places on those parts of the signal with the highest amplitude. The particular shape of the weighting function is somewhat arbitrary. However, its general aspects were chosen to impart some human behavior onto the algorithm. For example, the maximum-normalization conducted before weights are calculated ensures that the weighting function is scale-invariant. Furthermore, we added larger weights to values inside the measurement window without completely discarding the impact of values outside the measurement window. We also chose to square the normalized amplitude in order to reflect a non-linear relationship between amplitude and importance. The exact shape of this weighting function may be argued and optimized further.

### Results.

Regarding outcome measures, the MINSQ algorithm dominates the CORR algorithm in almost all indices inspected. It has better reliability, homogeneity, validity and is more robust to the impact of measurement windows. This provides evidence towards the argument that the MINSQ algorithm presents the better choice if one is limited to the application of just one algorithm. [Something about correlational methods having problems?, brunelli2009template talks about broad peaks in multiclass pattern recognition]

## Limitations

We chose not to implement a parameter shifting the entire template along the x-axis. Thus, in our algorithm the only way latency can be shifted is by also scaling the entire component. Later peaks thus necessitate broader components. This could be changed by introducing a parameter shifting the template without scaling it. However, this would also move the amplitude at 0 ms to some other timepoint. As the origin is of special importance in ERP research, we decided against this shifting parameter. It is the only fixpoint resulting from the averaging and baselining procedures. Thus, we chose not to disturb this property. Future work may investigate the impact the introduction of this additional parameter in template transformations has on the template matching algorithm. We also limited the algorithm to linear transformations of the template but could easily extend it to include non-linear scaling as well. Non-linear scaling would enable the template transformations to capture the effect of some participants not displaying speed differences in early components (low scaling here), but showing slow late components (higher scaling here).

During manual inspection of the choice made by the algorithms, we observed that they struggle especially with classifying subject-level ERPs containing two distinct peaks [See figure]. Both algorithms will return a match that may even fit quite well, but minor differences in the size of the two peaks can lead to inconsistencies across conditions with the algorithm choosing the first peak in one and the second peak in the other condition. Other algorithms face the same challenge. Human researchers can inspect all different ERPs belonging to the same subject and introduce some stability into the extraction method. Algorithms don’t typically allow for the use of information of a previous ERP in the extraction procedure of the current ERP. During manual inspection of the choices of our algorithm, this can of course be compensated for by the human researcher.

### The present study.

The generalizability of our findings is limited by the data we analyzed here. We inspected a limited sample of participants, range of tasks and number of components inspected. Depending on the component of interest, the effectiveness of different algorithms can vary ([Kiesel et al., 2008](#ref-kiesel2008measurement)). We suspect that the effectiveness of all algorithms will decline when attempting to extract earlier components. The P3 is a broad, high-amplitude and isolated component. This renders it ideal for algorithmic approaches, as the influence of surrounding components is comparatively low and the measurement window quite easily specified. Especially area latency approaches should diminish in quality due to the less isolated component structure of earlier components ([Luck, 2014](#ref-luck2014introduction)).

## Future research

Future research should focus on applying template matching algorithms to earlier components. We also suggest simulating data to be able to quantify the algorithms ability to recover the true latency of a component. This work serves largely as a proof-of-concept. The algorithms presented here have yet to prove themselves in a larger variety of tasks, samples and for different ERP components.

We also suggest improving the optimization processes used during our algorithms. The function used to implement the optimization of the MINSQ does not consistently converge on the global optimum, which we compensated for by initializing five different starting points and testing the solutions for convergence. This could be improved upon further.

Currently, the algorithms aim to identify this global optimum representing the absolute best similarity between transformed template and signal. It may be advantageous to rather use a linear combination of the best percentile of transformations as the solution of the optimization process. Brunelli ([2009](#ref-brunelli2009template)) raised this issue in the context of multiclass pattern recognition. Correlation filters tend to result in broad peaks of optimality. We currently just choose the absolute peak and the algorithm returns the corresponding transformation parameters. Choosing the highest point in that peak is influenced by noise in the same manner as peak latency algorithms. Future research should investigate using a linear transformation, like a weighted average, to make use of, for example, the top 0.1% of optimal transformations.

Aside from improvements in the implementation of the algorithm and extensions of the algorithms to earlier components, we will also improve the user interface employed for manual inspection of the choices the algorithms made. Currently, that interface displays the matched template and informs the researcher about the latency and fit statistic this match would result in. We also display the choices a peak latency and an area latency algorithm would have made. The researcher can then either accept the matched result, choose a result of the older algorithms, manually specify the component latency or reject the ERP overall due to poor identifiability. We will aim to improve this by adding a slider controlling the transformation parameters, allowing the researcher to manually match the template to the subject-level ERP. The functionality of manual latency specification will also be improved by integrating already existing software like the Measurement Tool provided by ERPLab [REF].

# Conclusion

This work provides proof-of-concept showing that template matching algorithms using the grand average as a template can be feasibly used to extract P3 latencies. Latencies extracted by our algorithm correlate highly with values extracted by an expert human researcher across tasks and preprocessing steps. Our algorithm is superior to previous algorithms like peak latency and area latency regarding the correlation with manually extracted latencies, homogeneity and robustness to the influence of different measurement windows. Our algorithms shows slightly worse reliability estimates compared to area latency approaches. A main benefit of our approach is the ability to quantify the algorithm’s confidence in a particular solution via a fit statistic. This allows researchers to inspect only the subset of ERPs with the worst fits and thus correct potential measurement error of the algorithm in a time-efficient manner. Overall, the MINSQ algorithm displays better qualities than the CORR algorithm but also results in a higher number of missing values. We will aim to improve the implementation of our algorithms and test their ability to extract earlier ERP components. Overall, the results obtained here leave us optimistic regarding the applicability of this template matching approach. It will likely provide a more objective and efficient way to extract ERP latencies without greatly compromising psychometric quality.

# References

Anderson, J. R., Zhang, Q., Borst, J. P., & Walsh, M. M. (2016). The discovery of processing stages: Extension of Sternberg’s method. *Psychological Review*, *123*(5), 481.

Borst, J. P., & Anderson, J. R. (2015). The discovery of processing stages: Analyzing EEG data with hidden semi-Markov models. *NeuroImage*, *108*, 60–73.

Briechle, K., & Hanebeck, U. D. (2001). Template matching using fast normalized cross correlation. *Optical Pattern Recognition XII*, *4387*, 95–102.

Brunelli, R. (2009). *Template matching techniques in computer vision: Theory and practice*. John Wiley & Sons.

Brunelli, R., & Poggiot, T. (1997). Template matching: Matched spatial filters and beyond. *Pattern Recognition*, *30*(5), 751–768.

Clayson, P. E., Baldwin, S. A., & Larson, M. J. (2013). How does noise affect amplitude and latency measurement of event-related potentials (ERPs)? A methodological critique and simulation study. *Psychophysiology*, *50*(2), 174–186. <https://doi.org/10.1111/psyp.12001>

Goshtasby, A., Gage, S. H., & Bartholic, J. F. (1984). A two-stage cross correlation approach to template matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *3*, 374–378.

Kiesel, A., Miller, J., Jolicœur, P., & Brisson, B. (2008). Measurement of ERP latency differences: A comparison of single-participant and jackknife-based scoring methods. *Psychophysiology*, *45*(2), 250–274.

Li, Y., Ma, Z., Lu, W., & Li, Y. (2006). Automatic removal of the eye blink artifact from EEG using an ICA-based template matching approach. *Physiological Measurement*, *27*(4), 425.

Liesefeld, H. R. (2018). Estimating the timing of cognitive operations with MEG/EEG latency measures: A primer, a brief tutorial, and an implementation of various methods. *Frontiers in Neuroscience*, *12*, 765.

Luck, S. J. (2005). Ten simple rules for designing and interpreting ERP experiments. *Event-Related Potentials: A Methods Handbook*, *4*.

Luck, S. J. (2014). *An introduction to the event-related potential technique*. MIT press.

Mahalakshmi, T., Muthaiah, R., & Swaminathan, P. (2012). Image processing. *Research Journal of Applied Sciences, Engineering and Technology*, *4*(24), 5469–5473.

Miller, J., Patterson, T., & Ulrich, R. (1998). Jackknife-based method for measuring LRP onset latency differences. *Psychophysiology*, *35*(1), 99–115. <https://doi.org/10.1111/1469-8986.3510099>

Miyake, A., Friedman, N. P., Emerson, M. J., Witzki, A. H., Howerter, A., & Wager, T. D. (2000). The unity and diversity of executive functions and their contributions to complex “frontal lobe” tasks: A latent variable analysis. *Cognitive Psychology*, *41*(1), 49–100.

Sadus, K., Schubert, A.-L., Löffler, C., & Hagemann, D. (2023). *A multiverse study for extracting differences in P3 latencies between young and old adults*.

Schubert, A.-L., Löffler, C., Hagemann, D., & Sadus, K. (2023). How robust is the relationship between neural processing speed and cognitive abilities? *Psychophysiology*, *60*(2), e14165.

Smulders, F. T. Y. (2010). Simplifying jackknifing of ERPs and getting more out of it: Retrieving estimates of participants’ latencies. *Psychophysiology*, *47*(2), 387–392. <https://doi.org/10.1111/j.1469-8986.2009.00934.x>

Ulrich, R., & Miller, J. (2001). Using the jackknife-based scoring method for measuring LRP onset effects in factorial designs. *Psychophysiology*, *38*(5), 816–827.

William, F., Aygun, R., & Zhu, F. (2020). ERP template matching for EEG single trial classification. *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2876–2883.

# Appendix

# Talking about appendices

First-level headers create appendix-sections labelled A-Z. You can print tables here as well and refer to them in your main part. They will receive a prefix to their Table/Figure Number based on the appendix section they are in.

# Another section

this creates another appendix section