

The masked priming toolbox: an open-source MATLAB toolbox for masked priming researchers

Andrew D. Wilson · James Tresilian ·
Friederike Schlaghecken

© Psychonomic Society, Inc. 2010

Abstract The Masked Priming Toolbox is an open-source collection of MATLAB functions that utilizes the free third-party PsychToolbox-3 (PTB3; Brainard, *Spatial Vision*, 10, 433–436, 1997; Kleiner, Brainard & Pelli, *Perception*, 36, 2007; Pelli, *Spatial Vision*, 10, 437–442, 1997). It is designed to allow a researcher to run masked (and unmasked) priming experiments using a variety of response devices (including keyboards, graphics tablets and force transducers). Very little knowledge of MATLAB is required; experiments are generated by creating a text file with the required parameters, and raw and analyzed data are output to Excel (as well as MATLAB) files for further analysis. The toolbox implements a variety of stimuli for use as primes and targets, as well as a variety of masks. Timing, size, location, and orientation of stimuli are all parameterizable. The code is open-source and made available on the Web under a Creative Commons License.

Keywords Masked priming · MATLAB

Masked priming is a standard experimental paradigm to investigate early or automatic stages of ongoing cognitive

processes not (yet) available to conscious control. It is used, for example, to examine the extent to which semantic content is activated by words that have not even been consciously perceived, to investigate the mechanisms that convey information about and produce responses to emotional facial expressions, and to explore how even simple motor responses are influenced by environmental information presented below the threshold for awareness (for reviews, see, e.g., Hannula, Simons, & Cohen, 2005; Kinoshita & Lupker, 2003; Sumner & Husain, 2008; Van Den Bussche, Van Den Noortgate, & Reynvoet, 2009). The latter one—masked motor priming—is perhaps conceptually the most straight-forward of these, and will be used as an example case here. In a typical masked motor priming experiment, participants have to give a speeded manual response to a visual target stimulus, which is preceded by a briefly presented and subsequently masked prime stimulus. On any given trial, the prime can be a stimulus associated with the same response as the subsequent target ('compatible trial'), with a different response ('incompatible trial'), or with no response ('neutral trial'). Under certain conditions, the mask prevents conscious perception of the prime. However, both consciously perceived ('supraliminal') and not consciously perceived ('subliminal') primes can affect the subsequent motor response, often in systematically different ways (e.g., Eimer & Schlaghecken, 2002). Priming effects in masked motor priming manifest either as positive or negative compatibility effects (PCEs or NCEs). With PCEs, responses are facilitated by compatible and impaired by incompatible primes (e.g., Neumann & Klotz, 1994). With NCEs, the reverse is true (e.g., Eimer & Schlaghecken, 1998). The magnitude and direction (PCE or NCE) of the priming effects vary systematically with the exact physical stimulus characteristics (e.g., shape, size, contrast, location, and timing, composition and variability

The code is available by request, or on the web at http://www.fbs.leeds.ac.uk/staff/Wilson_A/MaskedPrimingToolbox.zip

A. D. Wilson (✉)
Centre for Sport and Exercise Sciences,
Institute of Membrane and Systems Biology, University of Leeds,
Leeds LS2 9JT, UK
e-mail: A.D.Wilson@leeds.ac.uk

J. Tresilian · F. Schlaghecken
Department of Psychology, University of Warwick,
Coventry CV4 7AL, UK

of the mask, etc., e.g., Eimer & Schlaghecken, 2002; Schlaghecken, Blagrove, & Maylor, 2008). The paradigm has therefore provided researchers with an invaluable tool for investigating the control of visually triggered conscious (voluntary) and nonconscious (automatic) response selection and execution.

In the standard visuo-motor masked prime paradigm (e.g., Eimer & Schlaghecken, 2002) primes and targets are open arrowheads pointing to the left of right ('<<' and '>>'), the mask is a rectangular array of randomly oriented lines, newly constructed on every trial to prevent perceptual learning (Schubö, Schlaghecken, & Meinecke, 2001), and participants have to give a speeded left- or right-hand button-press response. Obviously, the motor control demands in such a task are very limited. In order to compare control of such discrete response choices with the control of continuous movement parameters like movement direction and force, the paradigm needs to be altered to encompass continuous stimulus parameters (e.g., graded stimulus size or direction) and response devices capable of recording continuous movement parameters (e.g., graphics tablets or force transducers). Such requirements are not easily implemented with standard laboratory equipment, especially if high temporal and spatial precision for both stimulus presentation and response recording is necessary.

The present article is a brief introduction to the Masked Priming Toolbox (MPT), a platform which we developed to expand the type of response measures and stimuli to be used in motor priming experiments. However, the toolbox is not restricted to the types of stimuli implemented here, and can be adapted to display stimuli suitable for semantic priming, emotional priming, etc. The MPT is available online under a Creative Commons License: researchers are welcome to use and modify the code to suit their needs, and we simply request that any changes and additions are contributed to the Toolbox so others can also use them. The zip file contains the MATLAB.m files, example parameter files (in Excel format) and a detailed manual.

Software and hardware requirements

This toolbox is written in MATLAB, a software environment designed by the MathWorks primarily for engineering applications but that is seeing increasing use in psychology and the life sciences for data analysis. MATLAB, while powerful, is not designed with extensive support for careful control of computer graphics - the MTB therefore relies on the free third-party Psychtoolbox-3 (PTB3: Brainard, 1997; Kleiner, Brainard & Pelli, 2007; Pelli, 1997; <http://psychtoolbox.org>). This widely used toolbox provides a MATLAB-based interface to code written in Open-GL that

provides all the required low-level hardware control needed for visual psychophysics. The MPT has been tested under Windows XP. The PTB offers preliminary support for Vista and Windows 7, with some caveats; to the extent that the PTB functions correctly in these environments, the MPT should as well.

The MPT pre-draws stimuli using the PTB-3 texture features; this enables them to be easily drawn to any screen location and in any orientation with reliable control over timing (a critical feature of priming experiments). This functionality requires a suitable graphics card and computer; please refer to the PTB-3 Web site for up-to-date details.

The MPT includes support (via PTB-3) for reading data from a standard graphics tablet; this should work on any tablet that implements the standard WinTab specification. It also supports reading data from a specific force transducer (an Interface SMT-1-100N transducer, connected via an SY034-USB amplifier).

Key functions and files

ExperimentParameters.txt

The primary interface to the MPT is the text file containing the experiment parameters. The toolbox contains an Excel (.xls) format file with an example set of parameters: the values can be changed, although the structure of the file must be preserved when saving. Editing in Excel is advised to ensure this. The file must then be saved out to a tab-delimited text file. There are four sections (see the manual for more details):

1. *Experiment Parameters*: settings that affect the entire run of an experiment, e.g., Response Device;
2. *Mask Parameters*: size and mask element parameters;
3. *Splash Screen Parameters*: A splash screen is shown at the start of every block to demonstrate the various imperative stimuli and the required responses;
4. *Trial Parameters*: each row specifies one unique trial type, which is shown a set number of times per block (specified in the Experiment Parameters). The user must specify one of every trial type required.

test.m

This function runs through a given parameter file but with a forced 2-s pause between stimuli transitions; this enables the user to ensure your primes, masks, and targets are all displaying correctly (e.g., in the correct location on the monitor). It is recommended to do this for all experiments.

primingMain.m

Once the experiment has been specified in the parameters file, it can be run from the MATLAB command window by calling the function *primingMain* and passing it two arguments: the name of the parameter file, enclosed in single quotes (e.g., 'ExperimentParameters.txt') and a participant ID code (e.g., 'sub01'). This function reads the parameter file, runs the experiment, and handles any data analysis; data are then output to both Excel and MATLAB files using the participant ID code in the file name.

Stimulus types

Primes and targets

At present, the MPT implements a set of common masked motor priming stimuli, and draws these using individual lines so they can be easily resized. Stimuli currently implemented include triple lines, double arrows, brackets, crosses (with equal or unequal lines), rectangles and 'exploded' versions of the rectangles and cross (i.e., with the lines unjoined; shapes containing line junctions are often difficult to mask properly) (see Fig. 1 for examples).

These stimuli can be used as primes or targets, drawn anywhere on the screen (using pre-specified default locations, e.g., 'centre', or manually defined locations) for any duration (exact values limited only by the monitor's screen refresh rate). They can be resized by adjusting a gain, and oriented to any angle around the clock. New figures can be added by adding a new sub-function to *makeStimuli.m* along the lines of the pre-existing code and adding to the switch/case logic that calls the relevant sub-function. This allows for the MPT to be easily expanded to handle, for instance, visual stimuli like words or schematic faces.

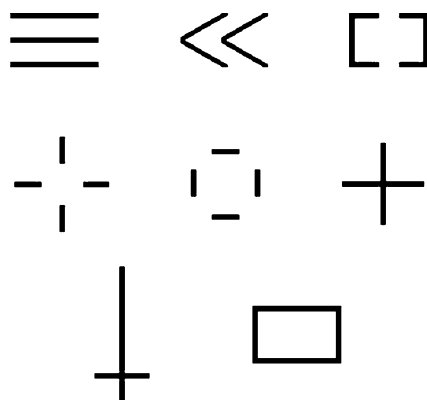


Fig. 1 Examples of the implemented stimuli that can serve as primes or targets. Any stimulus readily comprised of lines is straightforward to implement and add to the set

Masks

The MPT also implements three kinds of mask—'lines', 'random lines' and 'squares'. The 'lines' type resembles a mask commonly used in previous experiments (e.g., Eimer & Schlaghecken, 2002), containing 1/6 vertical lines, 1/6 horizontal lines, and 2/3 are oblique lines of various orientations. The 'random lines' type contains all random lines; 'squares' is composed of randomly sized squares (see Fig. 2).

The masks are created on the basis of a rectangular matrix. The algorithm subdivides a space of $xExtentPixels \times yExtentPixels$ into boxes of size $boxSize$. A random location is selected within each box, and an appropriately oriented line (or square) is drawn from this location. Line length and width are normally distributed, with the means and standard deviations as freely adjustable mask parameters. Varying these parameters manipulates mask density and composition. The result is a suitably constrained rectangular mask with jittered elements, generated anew on every trial to prevent perceptual learning of the mask (Schubö, Schlaghecken, & Meinecke, 2001). The mask can be drawn to any location, and for any duration allowed by the screen refresh rate. Masks generally need to be designed with the stimuli they are masking in mind; new masks, comprised of different elements, can be readily added by expanding the function *makeMask.m* in the same fashion as *makeStimuli.m*.

Response devices

The primary motivation for moving to MATLAB was to not expand the stimulus types, but to expand the response options. As mentioned above, the main aim was to enable investigation of continuous movement parameters as opposed to simple discrete movements.¹ The MPT currently reads data from (a) the standard keyboard, (b) any graphics tablet that implements the WinTab specification,² and (c) one specific type of force transducer (Interface SMT-1-100N transducer, connected via an SY034-USB amplifier; additional models should be straightforward to add using the PTB IOPort functionality demonstrated in the force transducer experiment MATLAB files).

Tablets Participants should be instructed to begin each trial in a start location (a 0.5-cm² box whose location is specified in the experiment parameters; the tablet should have this location indicated clearly). To ensure the

¹ So far, we have tested the use of a graphics tablet in some detail (Wilson, Tresilian & Schlaghecken 2010), and have piloted the use of the force transducer.

² Thanks to Mario Kleiner for his help implementing this functionality, which is now bundled with PTB-3.

Fig. 2 Examples of the implemented masks. The mask size and density can be altered by varying the mask parameters (larger ‘boxes’ mean fewer elements, and the normal distributions generating line lengths and widths can also be adjusted)



participant is ready and starting from a controlled location, stimulus presentation only begins once the pen has been held in the start box for 1 s. The response can then be any motion along the tablet surface. Most tablets can buffer around 500 ms of data; this buffer is used to store any movements made during stimulus presentation. Kinematics are output with $t = 0$ locked to the onset of the prime.

Some kinematic analyses have been implemented: the data are filtered, differentiated, and then various landmarks (movement onset, peak speed, etc.) are identified using an implementation of the algorithms in Teasdale, Bard, Fleury, Young, and Proteau (1993). Both raw and analyzed data (e.g., reaction time) are output to MATLAB and Excel files prefixed with the participant ID for further analysis.

The tablet data buffer can sometimes be empty if a pen movement has not been recorded correctly; this particular problem is difficult to catch reliably and will sometimes cause the experiment to end prematurely at the beginning of a block. Any remaining blocks can be run by altering the experiment parameter `nBlocks`, and the function `compileSessions.m` will subsequently copy the two separate data files into one result file. The kinematic analysis can then be run manually using the code in the Analysis Toolbox folder.

Force transducer In pilot work, we have trained participants to produce certain levels of force (expressed as a proportion of their maximum voluntary contraction; MVC) in response to various prime/target combinations. Force experiments therefore prompt the user to first ‘zero’ the transducer, then produce their MVC three times. The mean of these is then used to scale responses, display feedback, etc. If feedback is specified, the participant will see a bar whose height reflects the force they are producing, and a target for training (an oval in which they should place the top of the bar).

Other packages: ERTS & DMDX

The MPT is, of course, not the only available solution to the problems of producing carefully timed masked priming

experiments. ERTS (Berisoft Cooperation) is a simple and effective solution, but has become dated as DOS-based computers become impossible to find. A Windows-based system designed to cope with the timing problems caused by resource hungry operating systems is DMDX (Forster & Forster, 2003). This package has many strengths; however, we were motivated to develop the MPT within the MATLAB environment for several reasons. First, it enabled us to take full advantage of the extensive development, testing and support infrastructure for the Psychtoolbox (especially hardware support). This enabled us to operate under Windows without concern about timing control and rely on continued updates and technical support. Second, MATLAB is a powerful programming environment that is seeing more and more use in the life sciences. Many features we needed (such as kinematic data analysis for tablet data) are readily implementable in MATLAB, and adding functionality is simply a matter of some straightforward programming in a well supported high-level language. Third, DMDX is not open-source, and we wished to enable users to add needed functionality to suit their specific needs.

Conclusions

The MPT is designed to be a platform for vision and motor control researchers. All the basic functionality required to run masked and unmasked priming experiments has been implemented and tested, and many more advanced features (e.g., data analysis) have been implemented as well. At present, the focus of the toolbox is on the stimuli and masks required to run masked motor priming studies (e.g., Eimer & Schlaghecken, 2002). However, it was written to be indefinitely expandable to suit individual needs, with new stimuli types, analysis routines, etc., easily added, so that other priming paradigms like semantic or emotional priming can be implemented easily. Researchers are welcome to add functionality to the code themselves; we only request that you share useful functions with us so they can be incorporated into later versions. We hope the research community finds this toolbox a useful addition to their laboratories.

Acknowledgments This software was developed with support by ESRC Grant RES-000-22-2772 to FS & JT. Feature requests, new code and bug reports should be directed to ADW.

References

- Brainard, D. H. (1997). The psychophysics toolbox. *Spatial Vision*, 10, 433–436.
- Eimer, M., & Schlaghecken, F. (1998). Effects of masked stimuli on motor activation: Behavioral and electrophysiological evidence. *Journal of Experimental Psychology: Human Perception and Performance*, 24, 1737–1747.
- Eimer, M., & Schlaghecken, F. (2002). Links between conscious awareness and response inhibition: Evidence from masked priming. *Psychonomic Bulletin & Review*, 9, 514–520.
- Forster, K. I., & Forster, J. C. (2003). DMDX: A Windows display program with millisecond accuracy. *Behavior Research Methods*, 35(1), 116–124.
- Hannula, D. E., Simons, D. J., & Cohen, N. J. (2005). Imaging implicit perception: Promises and pitfalls. *Nature Reviews Neuroscience*, 6, 247–255.
- Kinoshita, S., & Lupker, S. J. (2003). *Masked priming: The state of the art*. NY: Psychology Press.
- Kleiner, M., Brainard, D., & Pelli, D. (2007). What's new in Psychtoolbox-3? *Perception*, 36, ECVF Abstract Supplement.
- Neumann, O., & Klotz, W. (1994). Motor responses to nonreportable, masked stimuli: Where is the limit of direct parameter specification? In C. Umiltà & M. Moskvitch (Eds.), *Attention and performance XV* (pp. 123–150). Cambridge, MA: MIT Press.
- Pelli, D. G. (1997). The VideoToolbox software for visual psychophysics: Transforming numbers into movies. *Spatial Vision*, 10, 437–442.
- Schlaghecken, F., Blagrove, E., & Maylor, E. A. (2008). No difference between conscious and nonconscious visuomotor control: Evidence from perceptual learning in the masked prime task. *Consciousness and Cognition*, 17, 84–93.
- Schubö, A., Schlaghecken, F., & Meinecke, C. (2001). Learning to ignore the mask in texture segmentation tasks. *Journal of Experimental Psychology: Human Perception and Performance*, 27, 919–931.
- Sumner, P., & Husain, M. (2008). At the edge of consciousness: Automatic motor activation and voluntary control. *Neuroscientist*, 14, 474–486.
- Teasdale, N., Bard, C., Fleury, M., Young, D. E., & Proteau, L. (1993). Determining movement onsets from temporal series. *Journal of Motor Behavior*, 25(2), 97–106.
- Van den Bussche, E., Van den Noortgate, W., & Reynvoet, B. (2009). Mechanisms of masked priming: A meta-analysis. *Psychological Bulletin*, 135, 452–477.
- Wilson, A. D., Tresilian, J. R., & Schlaghecken, F. (2010). Continuous priming effects on discrete response choices. *Brain & Cognition*, 74(2), 152–159.