A Computational Model of Task-Dependent Influences on Eye Position

http://ilab.usc.edu/rjpeters/pubs/2006_VSS_Peters.pdf

Robert J. Peters and Laurent Itti
University of Southern California, Computer Science, Neuroscience

Supported by the National Geospatial-Intelligence Agency (NGA)
and the Intelligence Community (IC) Postdoctoral Research Fellowship Program

introduction

We use ***fully-computational, autonomous*** models to predict eye movements in a ***dynamic and interactive*** visual task with ***naturalistic stimuli***.

We move beyond purely stimulus-driven bottom-up models, and introduce a simple model that captures task-dependent top-down influences on eye movements.

Previous studies have either relied on qualitative/descriptive models, or have not used naturalistic interactive stimuli.

	MODEL TYPE	
STIMULUS TYPE	qualitative	quantitative

static, artificial *(gabor patches, search arrays)*	Treisman & Gelade 1980;
	Wolfe & Horowitz 2004
		Rao et al 2002;
		Najemnik & Geisler 2005;
		Navalpakkam & Itti 2006

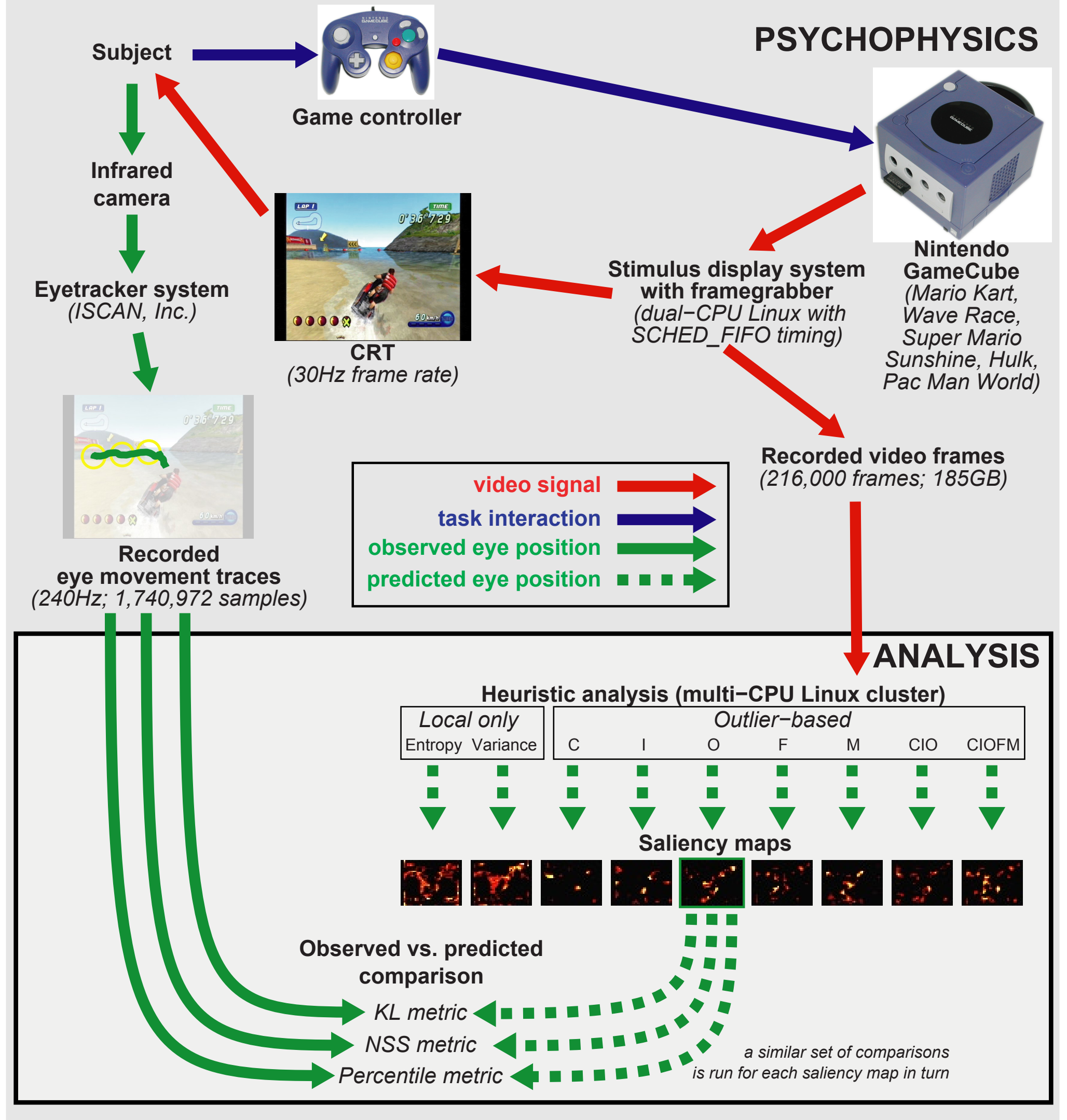
static, natural *(photographs)*	Yarbus 1967;
	Parker 1978;
	Mannan, Ruddock & Wooding 1997;
	Rayner 1998;
	Voge 1999
		Privitera & Stark 1998;
		Reinagel & Zador 1999;
		Parkhurst, Law & Niebur 2002;
		Torralba 2003;
		Peters et al 2005;
		Navalpakkam & Itti 2005;
		Pomplun 2006

dynamic, natural *(movies, cartoons)*	Tosi, Mecacci & Pasqual 1997;
	May, Dean & Barnard 2003;
	Peli, Goldstein & Woods 2005
		Carmi & Itti 2004;
		Itti & Baldi 2005

interactive, natural *(video games, flying or driving simulators, virtual reality, actual reality)*	Land, Mennie & Rusted 1999;
	Land & Hayhoe 2001;
	Hayhoe et al 2002;
	Hayhoe et al 2003
		this study

overall method

- eye movements recorded while subjects play video games
- eye movements compared with model predictions

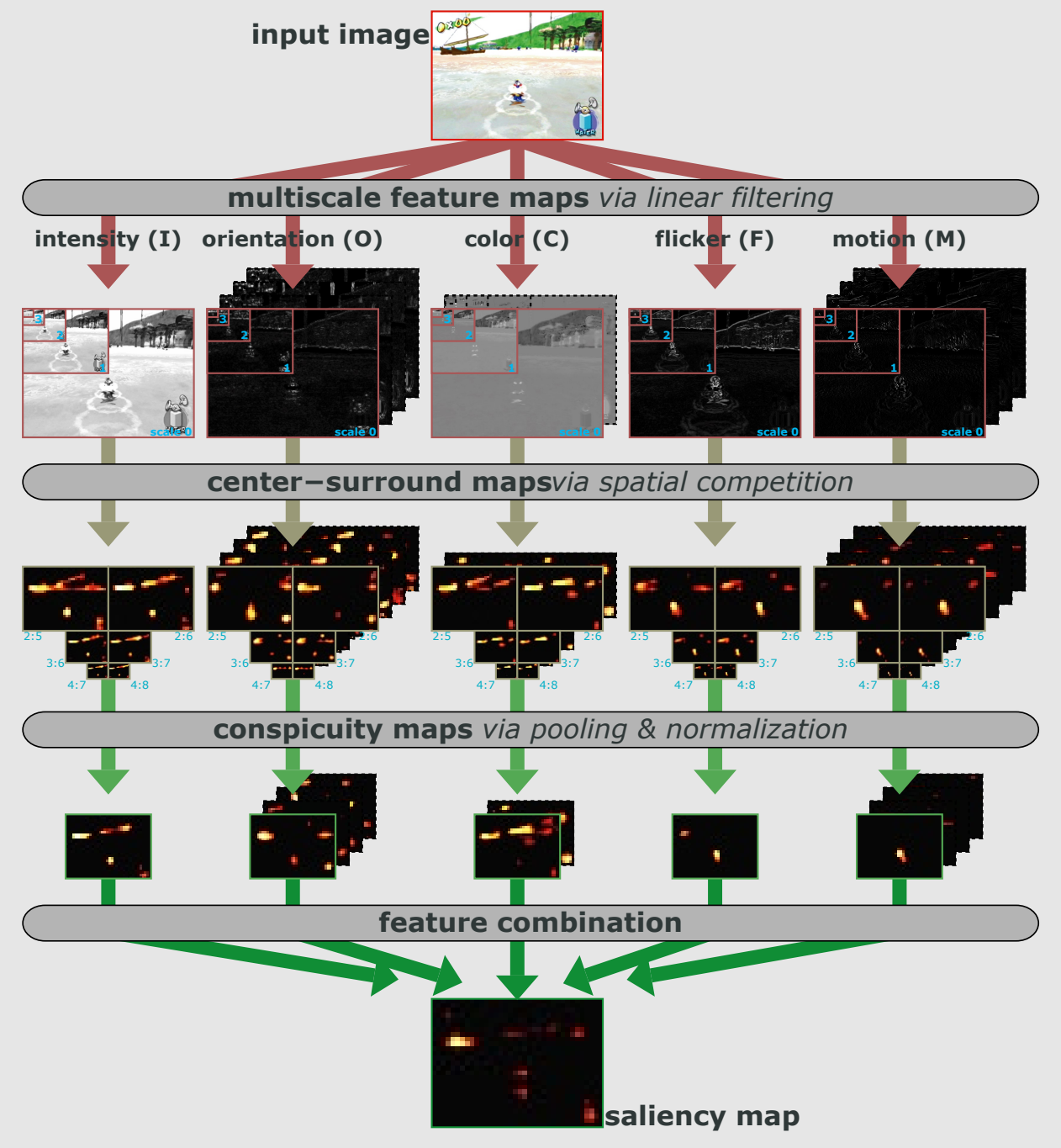

PSYCHOPHYSICS
Subject
Game controller
Infrared camera
Eyetracker system (ISCAN, Inc.)
CRT (30Hz frame rate)
Stimulus display system with framegrabber (dual−CPU Linux with SCHED_FIFO timing)
Nintendo GameCube (Mario Kart, Wave Race, Super Mario Sunshine, Hulk, Pac Man World)
Recorded eye movement traces (240Hz; 1,740,972 samples)
Recorded video frames (216,000 frames; 185GB)
video signal
task interaction
observed eye position
predicted eye position
ANALYSIS
Heuristic analysis (multi−CPU Linux cluster)
Local only
Entropy
Variance
Outlier−based
C
I
O
F
M
CIO
CIOFM
Saliency maps
Observed vs. predicted comparison
KL metric
NSS metric
Percentile metric
a similar set of comparisons is run for each saliency map in turn

videogame stimuli

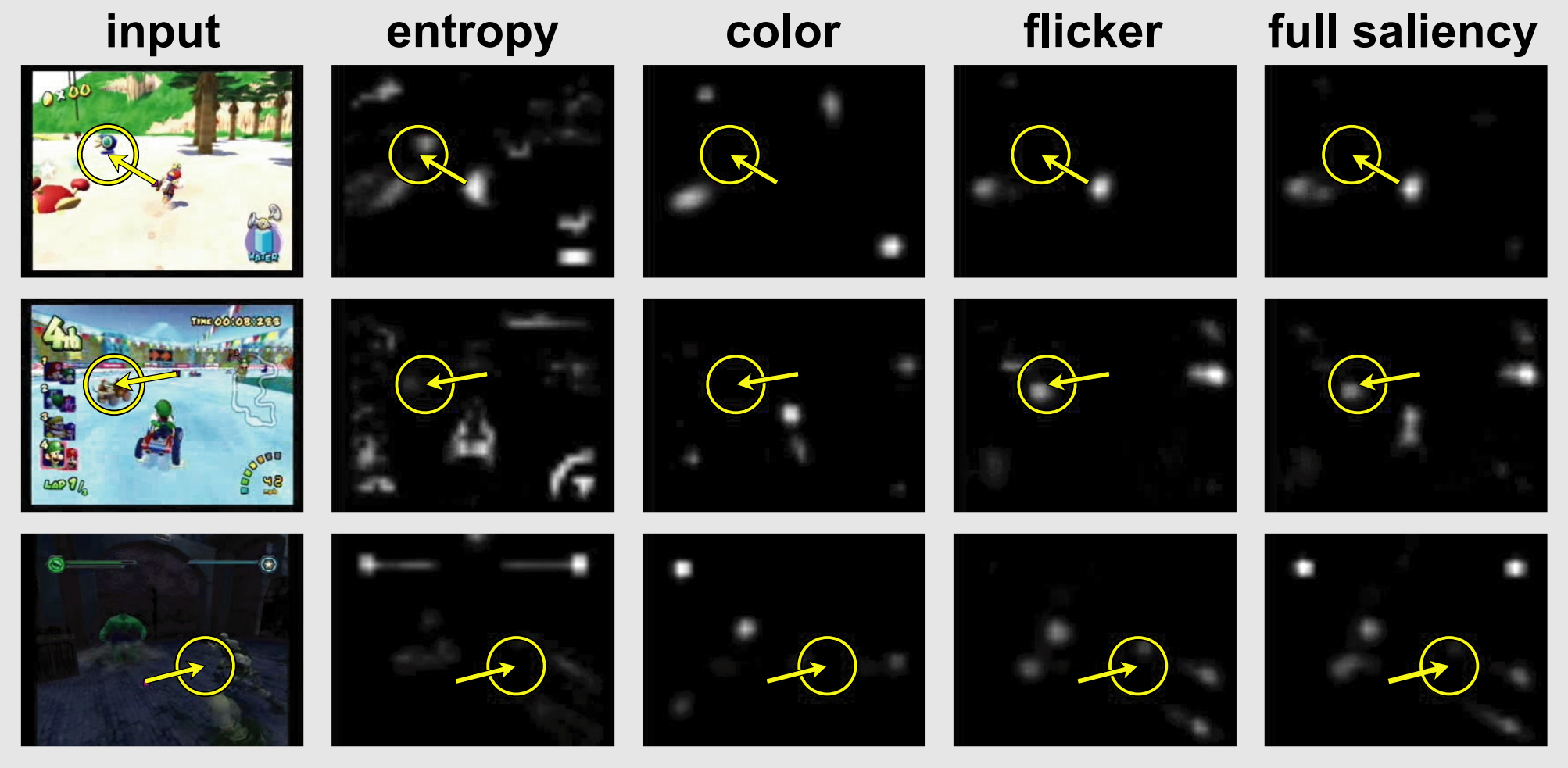
- Nintendo GameCube games
- 24 sessions, 5 minutes each

bottom-up model

- input processed through as many as 5 multi-scale feature channels
- different versions of the model include different combinations of features
- ***salience is a global property***: outliers detected through coarse global competition


input image
multiscale feature maps via linear filtering
intensity (I)
orientation (O)
color (C)
flicker (F)
motion (M)
center−surround maps via spatial competition
conspicuity maps via pooling & normalization
feature combination
saliency map

- sample frames illustrate the output of the models
- for illustration, these frames are selected at the time when a saccade is just beginning
- yellow arrows indicate the saccade path

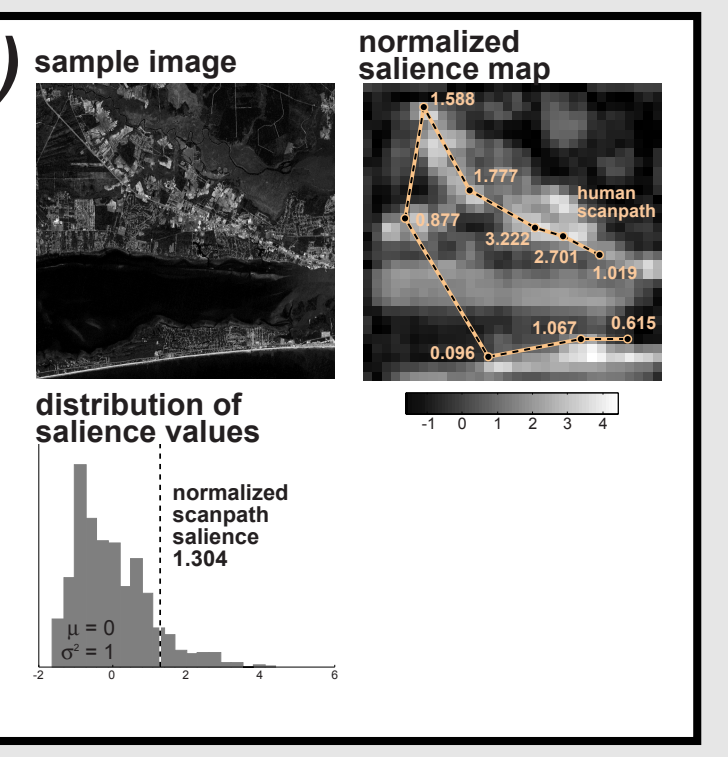


metrics: model scoring

- three different metrics used to test how well the models predict human eye movements

normalized scanpath salience (NSS)

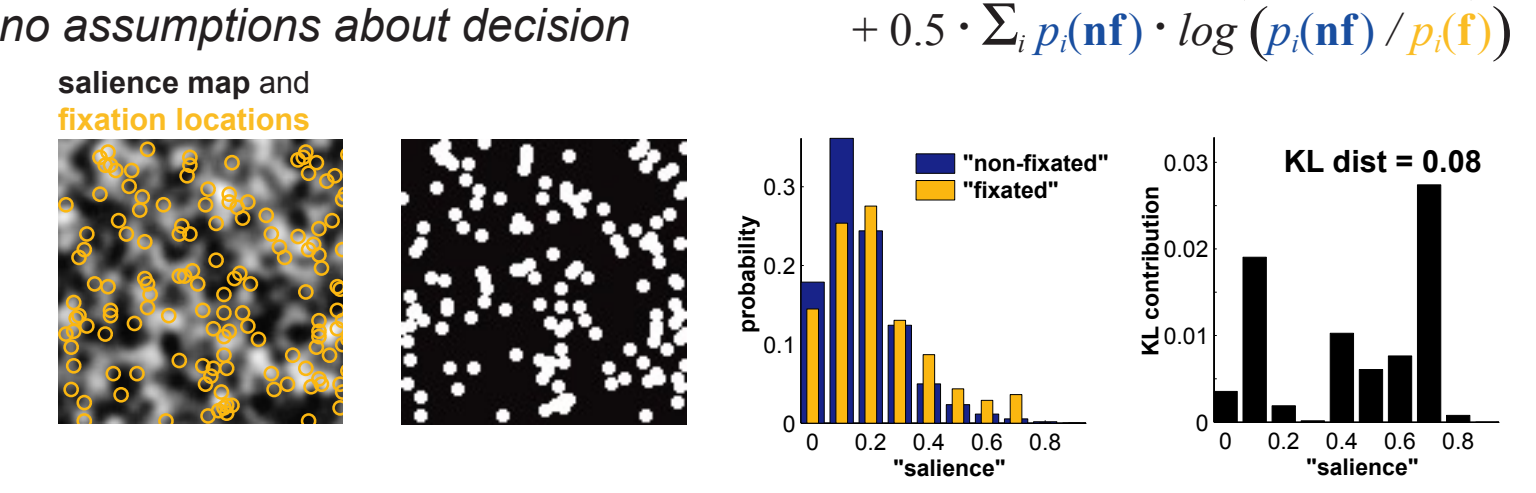
- each salience map (or control map) is normalized to have mean=0 and stdev=1
- human scanpath is overlaid on normalized salience map
- normalized salience value is extracted at each fixation location
- these values are summed to give the normalized scanpath salience (NSS)
- NSS can be compared with the distribution of random salience values



Kullback-Leibler (KL) distance

- how well can fixated and non-fixated locations be distinguished, *with no assumptions about decision criteria?*

$$KL = 0.5 \cdot \sum_i p_i(\mathbf{f}) \cdot \log\left(p_i(\mathbf{f}) / p_i(\mathbf{nf})\right) + 0.5 \cdot \sum_i p_i(\mathbf{nf}) \cdot \log\left(p_i(\mathbf{nf}) / p_i(\mathbf{f})\right)$$

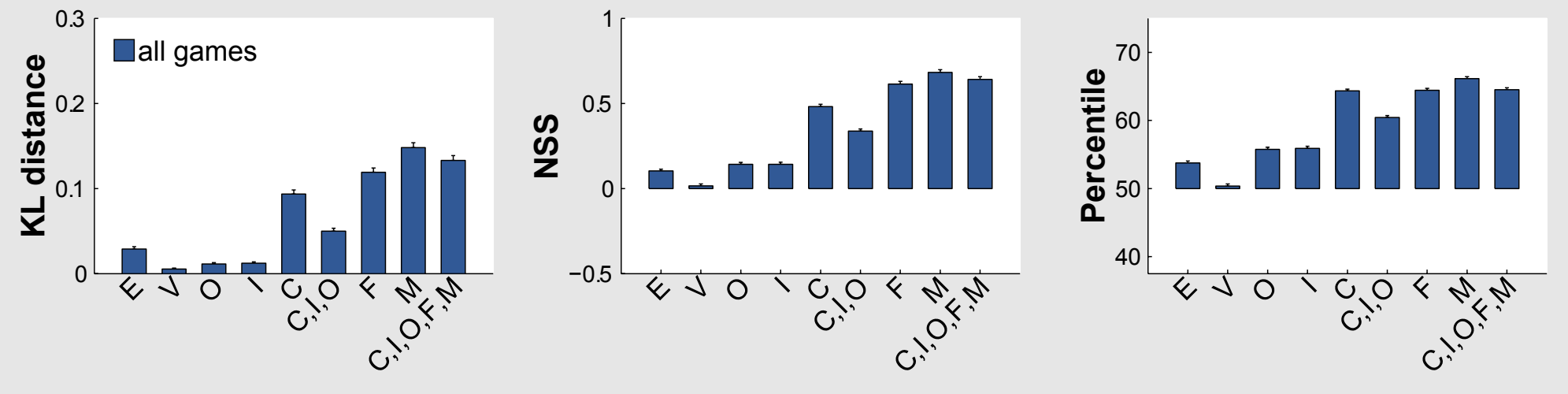


percentile

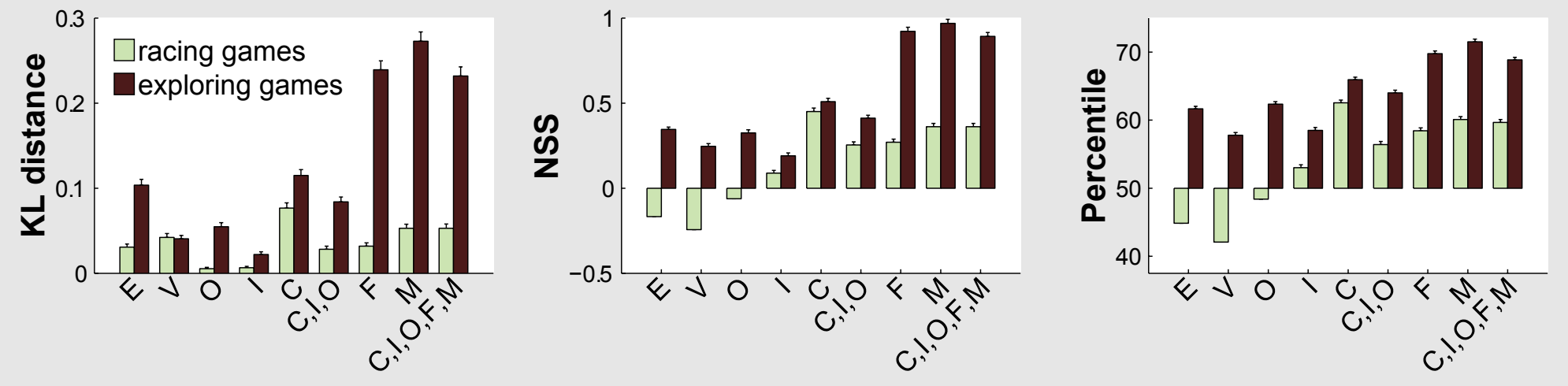
- within each salience map, what fraction of the values is exceeded by the value at the location of the human eye position?

bottom-up results

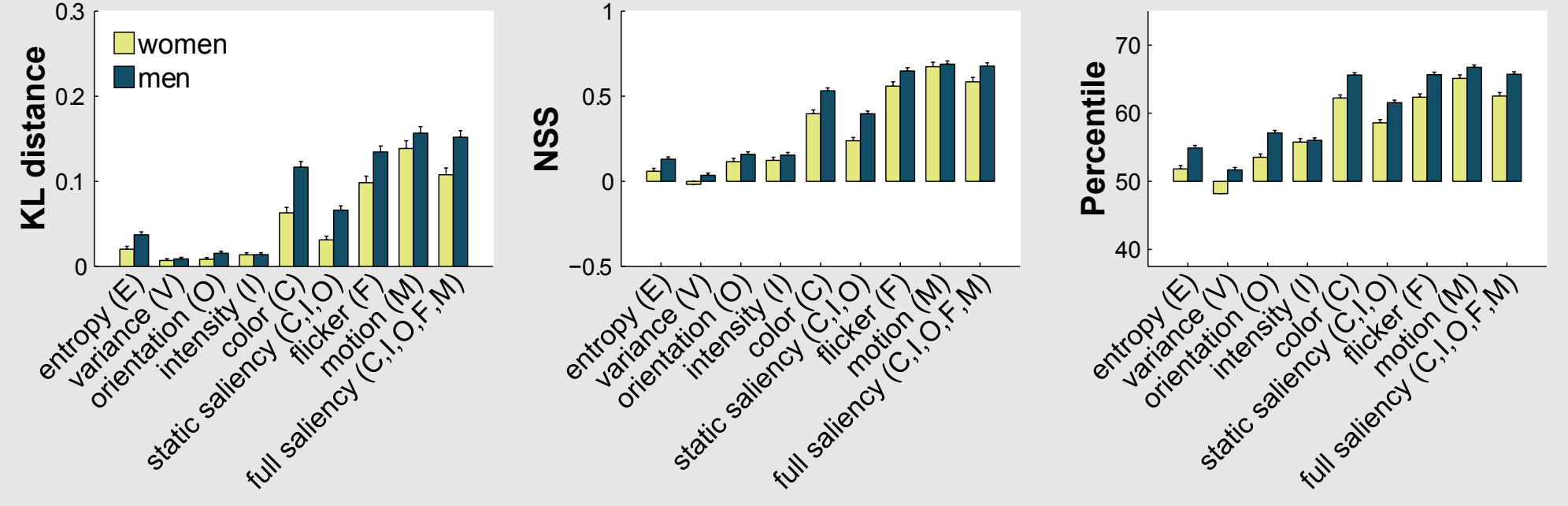
- models score significantly above chance
- dynamic, global features work best
- but, scores in this *interactive task* are lower than in previous *passive-viewing* experiments



- models score better for *exploring* than for *racing* games
- static color (C) is best for racing games
- dynamic motion (M) is best for exploring games

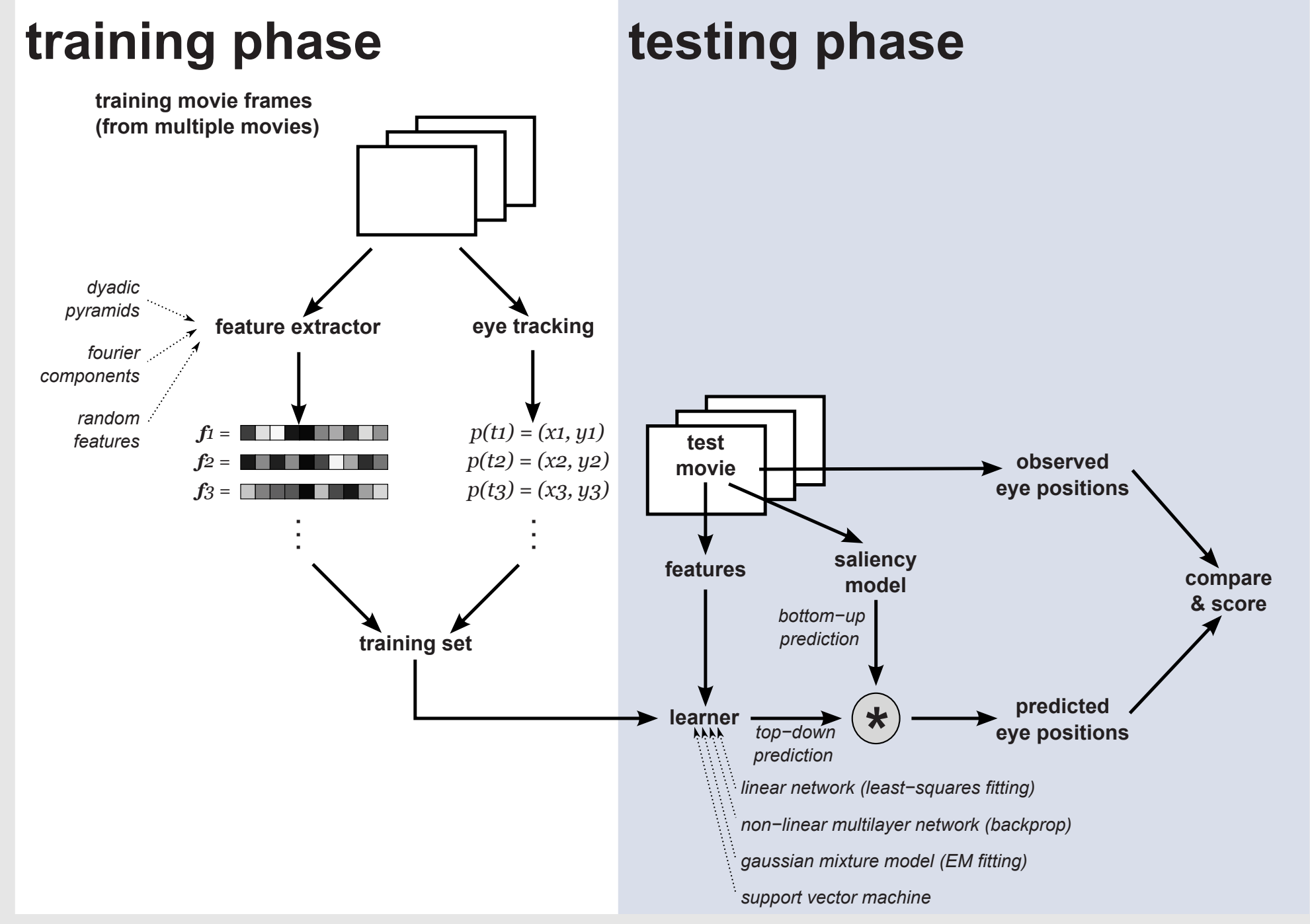


- men were more bottom-up driven than were women (but note that N is small: 3 men, 2 women)



top-down model

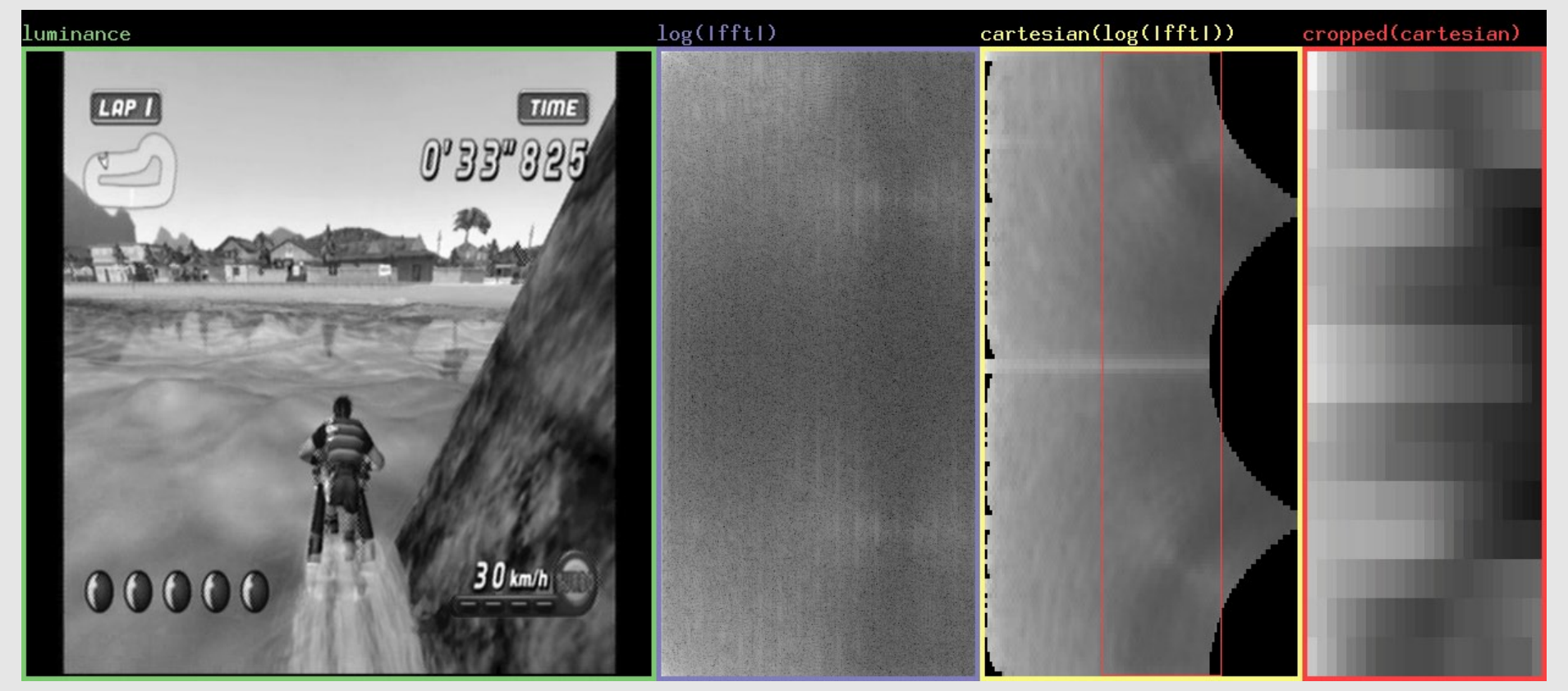
- model learns to associate ***image "gist" signatures*** with corresponding ***eye position density maps***
- in testing, a leave-one-out approach is used: for each test clip, the remaining 23 clips are used for training
- therefore, the model must be able to ***generalize across game types*** in order to successfully predict eye positions



feature extraction

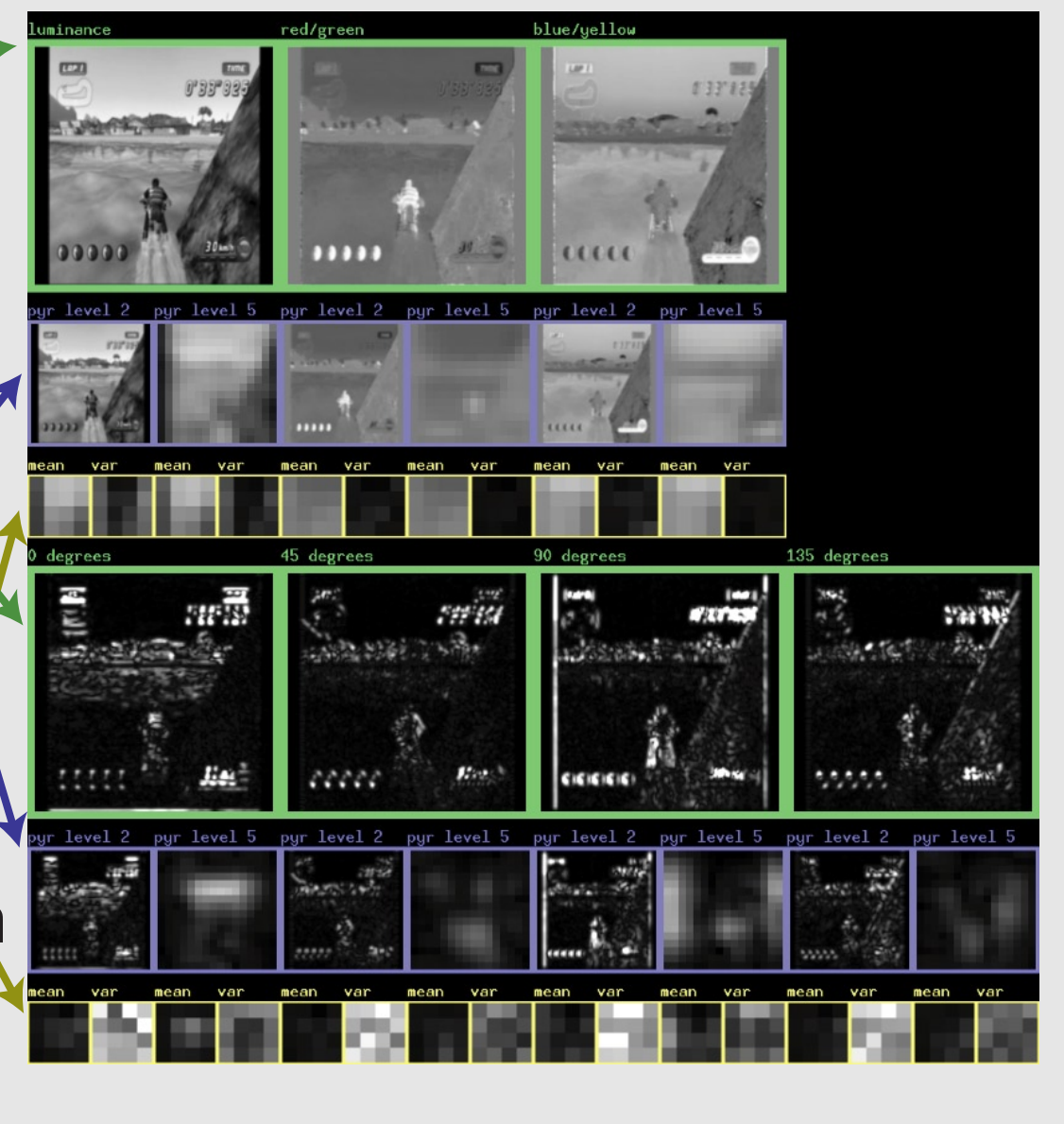
fourier features

- 384 features from the fourier transform
- fft log-magnitude converted to cartesian (θ,ω) space
- sample at 24x16 (θ,ω) locations



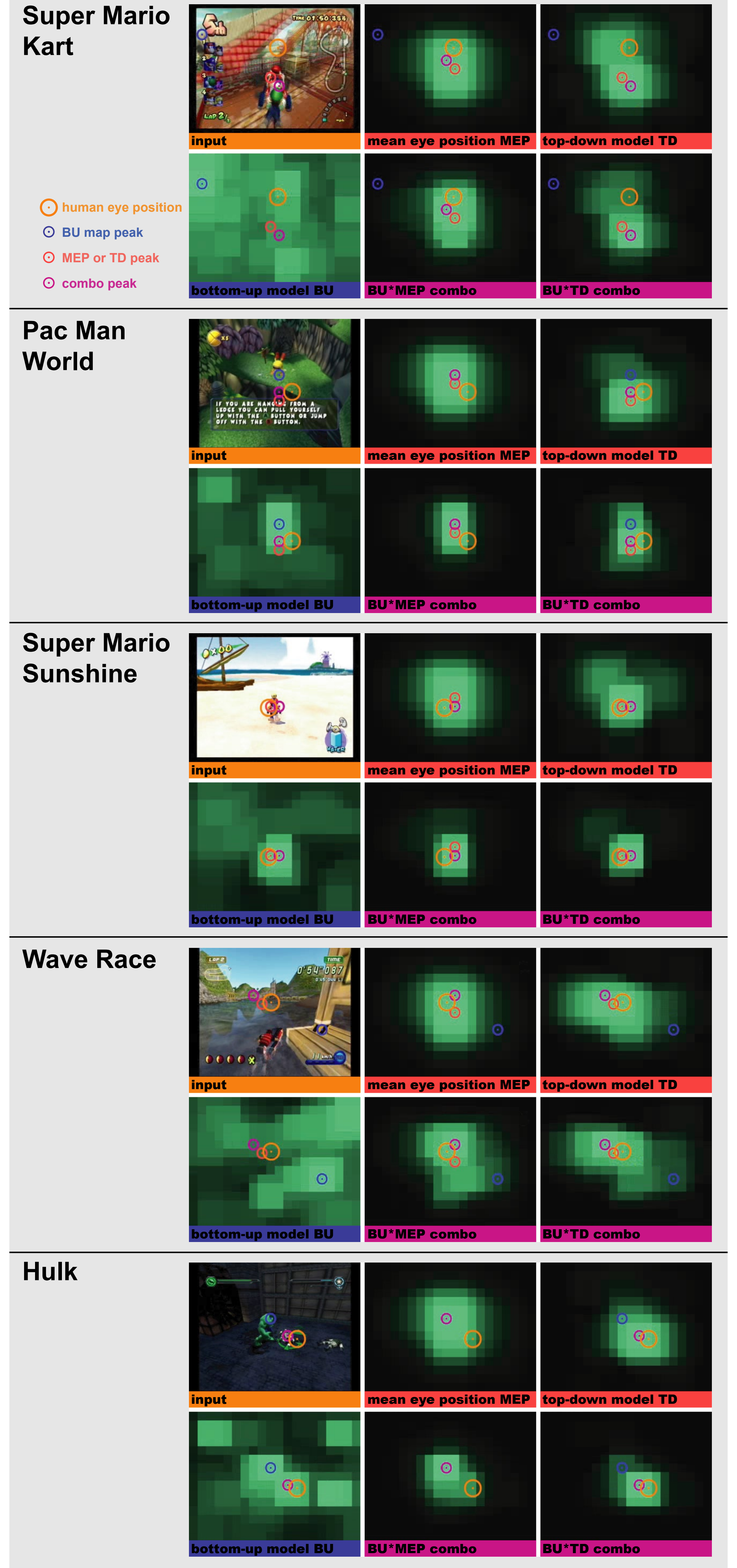
dyadic pyramid features

- 7 pyramids
  - 1 luminance
  - 2 color
  - 4 orientation
- 2 scales per pyramid
  - coarse
  - fine
- 32 features per scale
  - 4x4 array of local mean
  - 4x4 array of local var.



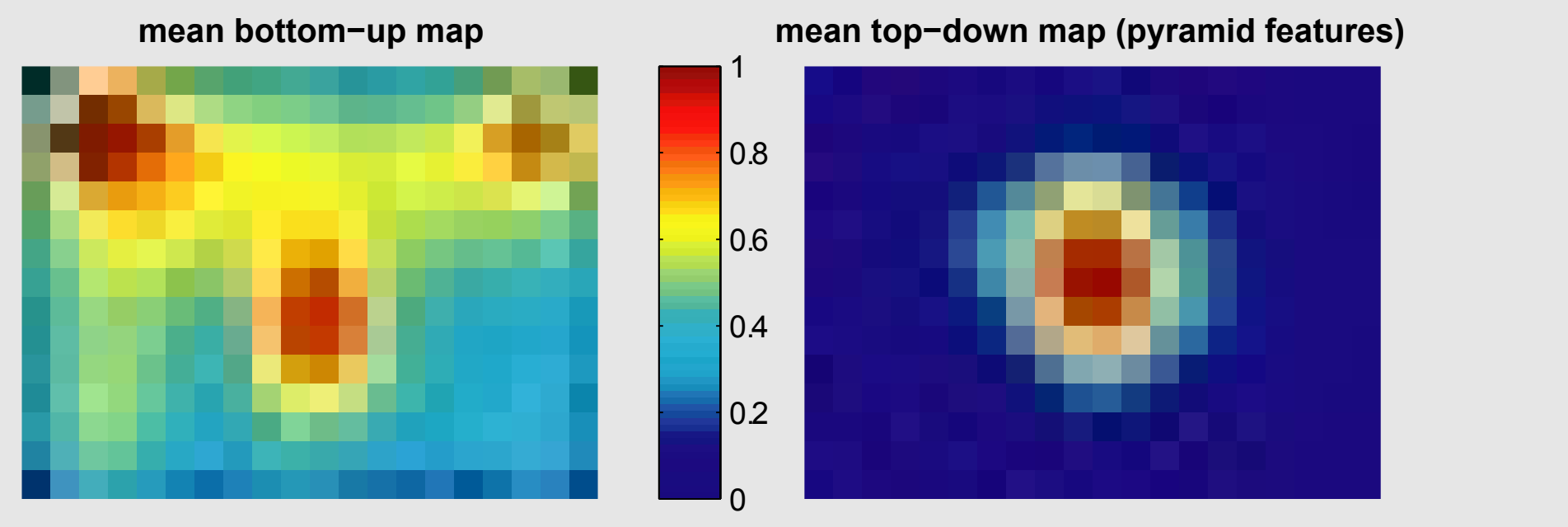
top-down predictions

- sample frames illustrating output of
  - **bottom-up (BU)** model alone
  - **mean eye position (MEP)** (a control condition)
  - **BU combined with MEP**
  - **top-down model (TD)** based on pyramid features
  - **BU combined with TD**

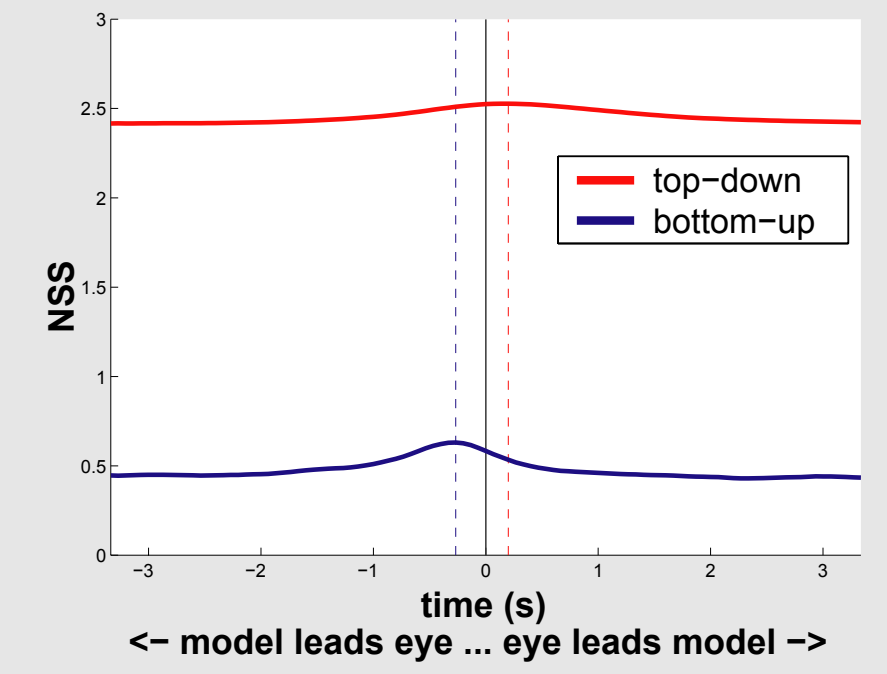


top-down results

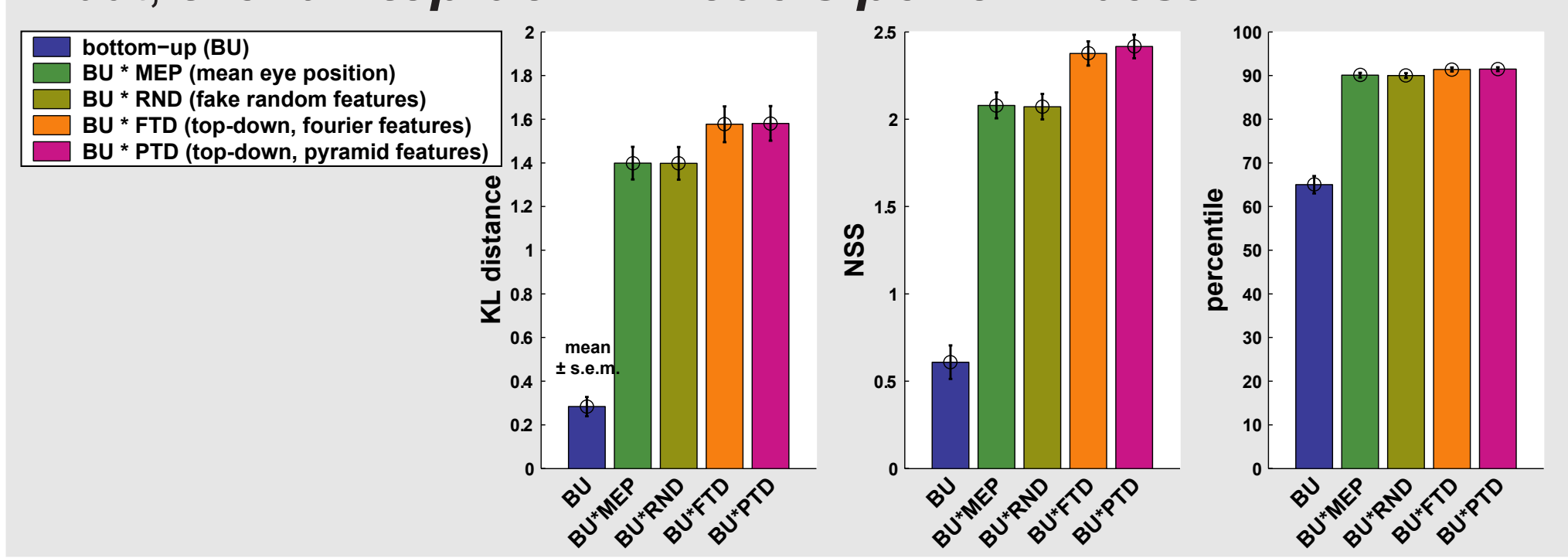
- average bottom-up and top-down maps across all frames
- bottom-up map reflects activity in the screen corners (game score, time counter, etc.) that is largely ignored by observers



- the bottom-up model is best predictive of eye position with about ~250ms delay
- the top-down model slightly lags behind eye position (due to temporal averaging in the model)



- a simple mean-eye-position control improves upon purely bottom-up performance by a factor of 3-4x
- but, ***the full top-down models perform best***



summary

- Goal was to explain gaze behavior with fully-computational models that don't know about "objects" or "actions"
- Purely bottom-up models are able to predict eye position significantly better than chance, but not as well as in passive-viewing conditions
- So, some other factors must be influencing eye position in our interactive task
- Introduced a simple model for learning top-down, task-dependent influences on eye position
- Combining bottom-up and top-down mechanisms leads to significantly improved eye position prediction
- This kind of model could be used in autonomous machine vision situations, such as interactive virtual environments