# Pyscanpath: A Framework To Load And Compare Scanpath Prediction Models On The Fly

Matteo Castagna
Università degli Studi di Milano
Department of Computer Science
Registration number: 27366A
Email: matteo.castagna2@studenti.unimi.it

***Abstract*—Here is proposed pyscanpath, a framework that wraps several models to generate human scanpaths and metrics to evaluate the predicted scanpaths with respect to a ground truth.**

## I. Introduction

Humans gather visual information about their surroundings using their eyes. We can collect high-density information only for small central area of our field of view. High visual acuity is restricted to the fovea, the small circular region (about 1.5 mm in diameter) in the central retina that is densely packed with cone photo receptors. Eye movements can direct the fovea to new objects of interest (foveation) or compensate for disturbances that cause the fovea to be displaced from a target already being attended to[1]. On this evidence, we direct our gaze to whatever might be the most relevant thing in the scene. How humans choose where to look attracted a lot of researchers in the last decades. The Scanpath Theory was defined by Noton and Stark [2] and become highly relevant in understanding human eye movements. Scanpaths are repetitive sequences of fixations and saccades that occur upon re-exposure to a visual stimulus, individuals who recognize a previously seen scene follow a scanpath similar to the one resulting from their initial viewing. In later works [3] a more relaxed scanpath theory, called "scanpath routines", suggests that humans rarely come across the same visual stimuli twice. Therefore, eye movements of individuals can rather be said to be similar between viewings of scenes or images from the same stimulus class.

Eye movements are an overt manifestation of the decision processes underlying attention. Evidence from experimental eye-tracking studies supports that internal cognitive structures control eye-movements and also the perception process itself. During the perceptual process, foveations enable the verification and adaptation of sub-features of cognitive structures. Based on these assumptions, human visual perception is seen mainly as a top-down process [4]. More generally, the deployment of attention to a particular location in space or to a particular feature or object can occur either by virtue of a stimulus physical salience (involuntary or bottom-up attention) or according to internal, behavioral goals (voluntary or top-down attention).

The studying of scanpaths is becoming more and more popular, as a tool for analysing visual perception and attention shift, scanpaths are being used to asses user interfaces [5], identify how individuals judge faces [3], studying facial emotion processing and impairments in interpersonal communication in patience with schizophrenia and ADHD [6] [7], studying threat conditioning [8] and studying cognitive processes during professional tasks, such as assessing learning-relevant student characteristics [9].

## II. Scanpath prediction models overview

In the following, is given an overview on the categories of existing scanpath prediction models. Models of scanpaths prediction can be classified based on where they take their inspiration from e.g., biology or statistics, etc.[10]. Four main classes can be identified.

### A. Biologically inspired models

Many models take their inspiration from results in neuroscience and vision science. One of the first ever developed scanpath prediction model is the one from Itti-Koch [11] which builds a saliency map based on top down attention and models fixations through a winner take all (WTA) network. Nowadays it is mainly used to predict saliency maps. Latest models, such as SceneWalk [12] and MASC [13], implement attention mechanisms together with saliency and inhibition-of-return from the seminal Itti-Koch model; the former use leaky memory to control the re-inspection of target regions while the latter applies results from neuro physiology to saccade prediction.

### B. Statistically inspired models

Statistically inspired models is a large class of models that don't take their inspiration from biology, but instead try to reproduce certain statistical properties of human scanpaths. Brockmann et al. [15] proposed a phenomenological model for the generation of human visual scanpaths where successions of saccadic eye movements are treated as realizations of a stochastic jump process in a salience field. CLE (Constrained Levy Exploration) [16] models saccades with a Levy flight, modulating the jump distribution with a saliency map. LeMeur15 [17] and LeMeur16 [18] similarly to the last models use a jump distribution but in this case it is approximated with non-parameterized methods. In SaccadicFlow [19] the jump distribution is a Gaussian which depends in a polynomial way from the previous fixations distribution.

### C. *Cognitively inspired models*

High level content such as objects and faces strongly attract attention, this raises the question of which cognitive effects affect gaze placement. The model by Liu et al. [20] models scanpaths based on three principal factors, namely low-level feature saliency, spatial position, and semantic content. Low-level feature saliency is formulated as transition probabilities between different image regions based on feature differences. The effect of spatial position on gaze shifts is modeled as a Levy flight with the shifts following a 2D Cauchy distribution. Hidden Markov Models (HMMs) are used to account for the semantic content. The IOR-ROI model [21] (Inhibition of Return - Region of Interest) integrates bottom-up features and semantic features extracted by convolutional neural networks. Then the integrated feature maps are fed into the IOR-ROI Long Short-Term Memory (LSTM). The IOR-ROI LSTM is a dual LSTM unit, capturing IOR dynamics and gaze shift behavior simultaneously.

### D. *Engineered models*

Engineered models are models which instead of implementing mechanisms taken from biological or statistical ideas, are deep learning based models that are fitted to the data. The SaltiNet [22] model uses a deep neural network to predict a spatiotemporal saliency volume which is then combined with a jump distribution to predict saccades. PathGAN [23] is a generative adversarial network architecture where the generator is a recurrent neural network predicting scanpaths the discriminator tries to discriminate the generated scanpaths from the ground truth human scanpaths. Finally, DeepGaze III [24], an extension for scanpath predictions of DeepGazeII [25] for spatial saliency modelling, uses deep features from the VGG19 network in a readout network of one by one convolutions to compute a spatial saliency map for an input image and encodes the previous two fixation locations in spatial feature maps. A final convolutional layer yield a distribution predicting the location of the next fixation in the scanpath.

## III. Models employed

In this project one model for each one of the categories reviewed in the last section is used. Here, the models employed are described more in details.

### A. *Itti-Koch*

The Itti-Koch model is a visual attention system inspired by the behavior and the neural architecture of the early primate visual system. The inspiration for the model comes by the remarkable ability of primates to interpret complex scenes in real time, despite the limited neuronal availability in such tasks, suggesting that intermediate and higher visual processes select a subset of the available sensory information before further processing [26]. Here multiscale image features are combined into a single topographical saliency map. Then a dynamical neural network selects attended locations in order of decreasing saliency.

The input to the model is provided as a static color image. A pyramid image is created using dyadic Gaussian pyramids, which progressively low-pass filter and subsample the input image. The final pyramid is composed by nine levels, or scales. Scale 0 is the input image at the original scale. Scale 8 is the image horizontally and vertically reduced by a factor of 1:256. Three main characteristics are considered in the computation of the saliency map: color, intensity and orientation. Red, green and blue color channels, four orientation channels corresponding to the angles $\theta \in \{0°, 45°, 90°, 135°\}$, built using Gabord filters, and an image intensity channel, have each one its own Gaussian pyramid. Features are built from each pyramid by means of "center-sorround" operation. Let's consider a "center" pixel at a fine scale $c \in \{2, 3, 4\}$ and the corresponding "surrounding" pixel, that is the same pixel obtained by interpolation to a coarse scale $s = c + \sigma$ with $\sigma \in \{3, 4\}$. The center-sorround, or across scale difference, between two maps is the point by point subtraction between the maps at the coarse and finer scales. Six features are obtained from each pyramid. Feature maps obtained from the same characteristic, color, orientation or intensity, are first normalized through local normalization and then combined by just summing them obtaining three so called conspicuity maps. The conspicuity maps are normalized again and then averaged obtaining the final saliency map.

At any given time, the maximum of the saliency map defines the most salient image location to which the focus of attention should be directed. The saliency map is modeled as a 2D layer of leaky integrate-and-fire (LIF) neurons at scale 4. When for a neuron the threshold is reached a spike is generated. The saliency map also feeds a winner take all neural network in which each neuron evolve independently from each other. When a neuron from the WTA reaches the threshold and fires several things happen: the FOA is shifted to the location of the winning neuron, all the neuron from the WTA reset and local inhibition is activated for the LIF neurons in the area and with the size of the new FOA. This allows for dynamical shift of the FOA.

### B. *CLE*

CLE models gaze shifts as realizations of a stochastic process in a saliency map representing the landscape upon which a constrained random walk is performed. Since the model takes advantage of both a deterministic mechanism, using the saliency map, and a random approach, it is based on the Langevin equation:

$$\frac{d}{dt}\vec{r} = -\vec{\nabla} V(\vec{r}) + \vec{\eta}, \tag{1}$$

where $V$ is a decreasing function of the saliency map (i.e. the deterministic part), while $\vec{\eta}$ is a stochastic vector. $\vec{\eta}$ is the stochastic component of the walk and is defined as follow:

$$\eta_x = l \cos(\alpha), \quad \eta_y = l \sin(\alpha), \tag{2}$$

where $l$ is the jump length obtained from a weighted Cauchy-Levy distribution and $\alpha$ is the direction of the flight, chosen

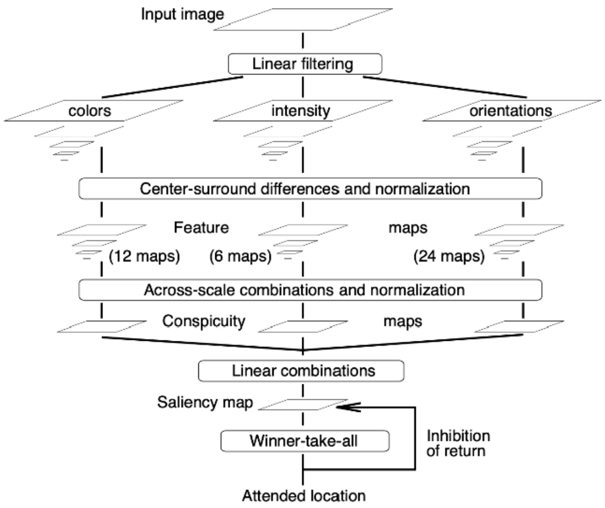

Fig. 1. General architecture of the Itti-Koch model

accordingly to a uniform distribution. Once the new gaze shift $\vec{r}_{new}$ is computed it goes through an acceptance process by the Metropolis algorithm. First the saliency gain $\Delta\hat{s}$ is computed as the difference between the weighted saliency at the previous and new fixation centers, then based on a parameter $T$ defining the randomness of the Metropolis algorithm, $\vec{r}_{new}$ is accepted with probability:

$$p(a|\vec{r}_{new}, \vec{r}) = \min\left\{1, \exp\left(\frac{\Delta\hat{s}}{T}\right)\right\} \tag{3}$$

### C. IOR-ROI

The IOR-ROI model tries to overcome some problems of the classical and biological based models that are: generating scanpath which depends on a single saliency map, predicting the next fixation without being conditioned on the previous one and ignoring high level semantic information from the image. The model is built on four major components: an image feature extractor, the region of interest (ROI) generator, the fixation duration prediction and a saliency guidance network. The image feature extractor module consists of a convolutional neural network, the VGG-19 pre-trained on the ImageNet dataset, which extract a feature map from the input image. An instance segmenter classifies and segment objects. The feature maps and the segmentation mask are fused with a $1\times1$ convolution creating a final feature map. Since its discovery, inhibition of return has been largely implemented in scanpath prediction, but IOR mechanism react differently to various visual features. To this end a IOR mechanism is built with a long short-term memory (LSTM) recurrent neural network, capable of modelling time series, exploiting as inputs both the feature map and the fixation history. LSTM network are a variant of recurrent neural network to deal with the vanishing gradient problem, still, classical LSTM are unable to preserve spatial information. Therefore, a ConvLSTM in which the fully connected components are substitute by convolution operations, is used. The IOR-LSTM takes in input the feature map from the previous step, the ROI map indicating the current region of interest by a 2D Gaussian blob, the fixation duration of the current fixation and the hidden state of the ROI-LSTM which encode attention history. In the map outputted by the IOR-LSTM, each channel represents the usage of a certain feature when focusing on the ROI, hence it can be considered as the inhibition strength of each feature. The inhibited feature map is concatenated with the hidden state of the ROI-LSTM by a convolved with a $5 \times 5$ convolution. Channel wise attention is computed by applying a softmax on fixated regions and context region and then the channels of the inhibited map are weighted. Finally the ROI-LSTM takes the weighted map. To select the next fixation the hidden state of the ROI-LSTM is flattened and fed to a Mixed Density Network (MDN) which predicts the parameters of a multimodal distribution. A set of $n$ examples is sampled from the predicted distribution. The sample which maximize the product between the probability of the sample itself to be drawn and the joint probability distribution of the saccade amplitude and orientation, is chosen as the next fixation point. Task related knowledge and saliency map predicted by a Saliency Guidance Network from the feature extracted by the VGG-19, are used to regularize the training by constraining the parameter space.

### D. DeepGazeIII

DeepGazeIII approaches the scanpath prediction problem using probabilistic generative modeling. The distribution of scanpaths is modeled given the image and the initial center fixation $p(f_1, f_2, f_3, \ldots, f_N|f_0, I)$. To make the problem more tractable, the scanpaths distribution is split:

$$p(f_1, f_2, f_3, \ldots, f_N|f_0, I) = \prod_{i=1}^{N} p(f_i|f_0, \ldots, f_i - 1, I). \tag{4}$$

This procedure agrees with evidence from neuroscience [28] which suggests that while fixating a point in an image, the brain selects where to saccade next by incorporating task, oculomotor biases, and memory. In other words, where we looked before influences where we might look next. Scanpaths are generated by the model by sampling from its distribution, making use of the chain rule decomposition. The first fixation $f_1$ is sampled from $p(f_1|f_0, I)$. Then it is used to sample the second fixation $f_2$ from $p(f_2|f_0, f_1, I)$ and so on until a scanpath of the desired length is obtained. DeepGazeIII takes the four most recent fixations made by the subject into account, the current fixation and the three previous fixations. The model receives as input an image and the scanpath history and outputs the conditional fixation distribution, which is a two-dimensional probability density and encodes where the model expects the subject to fixate next. The image is downscaled by a factor of 2 and processed with a convolutional neural network. By extracting the results of the activation functions of multiple layers of the convolutional network a deep representation of the image is created. The result is a map of 2048 channels. This map is sent to a spatial priority network, that is just a readout network which maps the incoming data to an higher dimensional space and then performs a linear transformation of the mapped data. The output of the spatial priority network is a single layer map, called spatial priority map. Parallel to the spatial priority network there is a scanpath network that processes the scanpath history. Each fixation that the model receives information from is encoded into three spatial feature maps encoding Euclidean distance and differences in fixation coordinates. The maps are merged through convolution. Finally the outputs from the two networks are combined through convolution and fed to fixation selection network, still a convolutional network. The output is blurred and normalized with a softmax to output a conditional fixation distribution:

$$p(f|f_{i-1}, \dots, f_{i-4}, I). \tag{5}$$

## IV. Scanpath metrics

In this project four metrics to compute scanpath similarity are used. None of the metrics take into account the duration of the scapath or the duration of each fixation, as not all the models employed are able to return those data. Here, the metrics implemented are reviewed. To make all the metrics comparable and to ensure their proportionality when computing the average in the testing phase, all the metrics lie in range between 0 and 1.

### A. Euclidean distance

Euclidean distance is one of the initial metrics that was used in comparing scanpaths. The distance can be calculated as the sum of the distances between fixation pairs each with two-dimensional Cartesian coordinates ($P_i^x$ , $P_i^y$):

$$D(P,Q) = \sum_{i=1}^{min(N,M)} \sqrt{(P_i^x - Q_i^x)^2 + (P_i^y - Q_i^y)^2}, \tag{6}$$

where $P$ and $Q$ are the two scanpaths considered and $N$ and $M$ their respective lengths.

### B. Mannan distance

The Mannan distance is a metric comparing scanpaths where the order of fixations is completely ignored. The Mannan distance compares the similarity between scanpaths by calculating the distance between each fixation in one scanpath and its nearest neighbour in the other scanpath. A similarity index ($Is$) represents the average linear distance between two scanpaths ($D_{P,Q}$), with randomized scanpaths having the same size ($D_r$):

$$Is = \left[1 - \frac{D_{P,Q}}{D_r}\right] \cdot 100, \tag{7}$$

$$D_{P,Q} = \frac{M \sum_{i=1}^{N} \min d_{P_i}^2 + N \sum_{j=1}^{M} \min d_{Q_j}^2}{2NM(W^2 + H^2)}, \tag{8}$$

where $P$ and $Q$ are the two scanpaths considered and $N$ and $M$ their respective lengths, $W$ and $H$ are respectively the width and the height of the image, $d_{P_i}^2$ is the distance between the i$^{th}$ fixation in the first scanpath and its nearest neighbor in the second one, and $d_{Q_j}^2$ is the distance between the j$^{th}$ fixation in the second scanpath and its nearest neighbor in the first one. These randomly generated scanpaths are used for weighting the sequence of real fixations, taking into account the fact that real scanpaths may convey a randomized component. The main drawbacks of this technique are: not taking into account the temporal order of fixation sequence and the Mannan distance is not tolerant to high variability between scanpaths, which makes it a quite bad metric for scanpaths which sizes are largely different. The similarity index range from 0, random scanpaths, to 1, totally similar scanpaths.

### C. Edit distance

The idea of the string edit metric is that a sequence of fixations can be translated into a sequence of symbols (numbers or letters) forming strings that are compared. This comparison is carried out by calculating a string edit distance (often called the Levenshtein distance) that gives a measure of the similarity of the strings. This technique was originally developed to account for the edit distance between two words, and the measured distance is the number of deletions, insertions or substitutions that are necessary for the two words to be identical. This metric takes as input two strings (coding AOIs) and computes the minimum number of edits needed to transform one string into the other. A cost is associated with each transformation and each character. The total cost is the edit distance between the two strings. When the cost is minimal, the similarity between the two strings is maximal. The advantage of the string edit technique is that it is easily computed and keeps the order of fixations. However, several drawbacks have to be underlined:

- since the string edit is based on a comparison of the sequence of fixations occurring in pre-defined AOIs, the question is how to define these AOIs. There are two ways: automatically gridded AOIs or content-based AOIs. In the former, each grid cell is assigned an alphabetic character. Following this, a series of fixations can be represented using a string of characters and the Levenshtein algorithm can be used to compare strings representing them;
- the string edit method is limited when certain AOIs have not been fixated so there is a good deal of missing data.

### D. Time delay embedding

Time delay embedding (TDE) tries to consider the ordering by creating consecutive subsamples in the scanpath. Scanpaths $P$ and $Q$ go through the same pre-processing step by breaking each into consecutive samples of a given length $K$. The final output comes from calculating minimum of pair-wise Euclidean distance between subsamples from each scanpath. The parameter $K$ plays an important role in handling spatial and ordinal noise.

## V. About the framework

Each one of the models in pyscanpath has its how module but the instantiation and the usage work the same for all the models. Let's consider, for example, the DeepGazeIII model, the class Deepgaze can be imported in the following way, `from pyscanpath.models.Deepgaze` `import Deepgaze` and an object from the class can be instantiated via the following function: `Deepgaze()`. The method `.getScanPath()` is used to generate a scanpath. This methods takes two inputs: the path of the image to predict the scanpath and optionally the number of fixations to predicts, which is ten by default. For Itti-Koch model and CLE, `.getScanPath()` takes additional arguments:

- Itti-Koch: `saliency_type` (default '01')
- CLE: `saliency_type` (default '98'), `tauV` (default 0.01), `numSampleLevy` (default 50), `dt` (default 0.1), `T` (default 25)

`saliency_type` is a string which define the type of saliency map to compute on which the scanpath will be predicted, it can be either '98' or '01'.

The `metrics` module contains the functions to compute distances or similarities measurements between scanpaths. Four metrics are implemented: euclidean distance, Mannan distance, edit distance and time delay embedding. Edit distance and time delay embedding implementations are taken from FixaTons[1].

Finally, pyscanpath supports the MIT1003 dataset, the CAT3000 dataset and the OSIE dataset, to which it accessed via pysaliency[2]. The function `getDatasetScanpath()` from the `datasets` module let access to stimuli and scanapth in the dataset by specifying four arguments:

- the name of the dataset to use, either 'MIT1003', 'CAT2000' or 'OSIE'
- the name of the folder which stores the dataset downloaded by 'pysaliency'
- the index of the image in the dataset to analyze
- the index of a subject

and returns: the path of the image is given the index, the image for which is give the index, the scanpath of the image and subject for which is given the index.

## VI. Testing

A brief comparison between the models has been done testing them on 100 stimuli drawn uniformly from the MIT1003 dataset, for each one of the 100 stimuli a random subject from th 15 available is picked. The test has been performed with a 2021 Apple M1 MacBook Pro with a 16 GB RAM and 10 Core on macOs Ventura 13.5.2. Computing the scanpaths for both the four models for one stimuli and one subject takes approximately 52 seconds, this factor lead to choose to use only 100 stimuli for testing due to time constraints. Results are presented in Tab. I

TABLE I
Average and standard deviation metrics computed over the the 100 uniformly distributed stimuli from the MIT1003 dataset for the 4 models implemented in the framework.

| Model | Euclidean | Mannan | Levenshtein | TDE |
|---|---|---|---|---|
| Itti-Koch | 0.63±0.10 | 0.20±0.16 | 0.93±0.08 | 0.82±0.03 |
| CLE | 0.63±0.10 | 0.32±0.21 | 0.93±0.08 | 0.82±0.03 |
| DeepGaze III | 0.63±0.10 | 0.35±0.20 | 0.89±0.10 | 0.82±0.03 |
| IOR-ROI | 0.27±0.83 | 0.33±0.22 | 0.90±0.09 | 0.93±0.02 |

## References

[1] Purves D, Augustine GJ, Fitzpatrick D, et al., editors. Neuroscience. 2nd edition. Sunderland (MA): Sinauer Associates; 2001. What Eye Movements Accomplish.

[2] Noton D, Stark L. Scanpaths in eye movements during pattern perception. Science. 1971 Jan 22;171(3968):308-11. doi: 10.1126/science.171.3968.308. PMID: 5538847

[3] Kanan C, Bseiso DN, Ray NA, Hsiao JH, Cottrell GW. Humans have idiosyncratic and task-specific scanpaths for judging faces. Vision Res. 2015 Mar;108:67-76. doi: 10.1016/j.visres.2015.01.013. Epub 2015 Jan 30. PMID: 25641371

[4] Lawrence W. Stark, Yun S. Choi, Experimental metaphysics: The scanpath as an epistemological mechanism, Advances in Psychology, North-Holland, Volume 116

[5] Le Meur, Olivier Baccino, Thierry. (2012). Methods for comparing scanpaths and saliency maps: Strengths and weaknesses. Behavior research methods. 10.3758/s13428-012-0226-9.

[6] Loughland, C. M., Williams, L. M., Gordon, E. (2002). Visual scanpaths to positive and negative facial emotions in an outpatient schizophrenia sample. Schizophrenia research, 55(1-2), 159–170. https://doi.org/10.1016/s0920-9964(01)00186-4

[7] Marsh, P. J., Williams, L. M. (2006). ADHD and schizophrenia phenomenology: visual scanpaths to emotional faces as a potential psychophysiological marker?. Neuroscience and biobehavioral reviews, 30(5), 651–665. https://doi.org/10.1016/j.neubiorev.2005.11.004

[8] Xia, Y., Melinscak, F., Bach, D. R. (2021). Saccadic scanpath length: an index for human threat conditioning. Behavior research methods, 53(4), 1426–1439. https://doi.org/10.3758/s13428-020-01490-5

[9] Kosel, Christian Holzberger, Doris Seidel, Tina. (2021). Identifying Expert and Novice Visual Scanpath Patterns and Their Relationship to Assessing Learning-Relevant Student Characteristics. Frontiers in Education. 5. 10.3389/feduc.2020.612175.

[10] Kümmerer, Matthias Bethge, Matthias. (2021). State-of-the-Art in Human Scanpath Prediction.

[11] L. Itti, C. Koch and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 11, pp. 1254-1259, Nov. 1998, doi: 10.1109/34.730558

[12] Engbert, R., Trukenbrod, H. A., Barthelmé, S., Wichmann, F. A. (2015). Spatial statistics and attentional dynamics in scene viewing. Journal of Vision, 15(1), Article 14. https://doi.org/10.1167/15.1.14

[13] Adeli H, Vitu F, Zelinsky GJ. A Model of the Superior Colliculus Predicts Fixation Locations during Scene Viewing and Visual Search. J Neurosci. 2017 Feb 8;37(6):1453-1467. doi: 10.1523/JNEUROSCI.0825-16.2016. Epub 2016 Dec 30. PMID: 28039373; PMCID: PMC6705681

[14] D. Zanca, S. Melacci and M. Gori, "Gravitational Laws of Focus of Attention," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 12, pp. 2983-2995, 1 Dec. 2020, doi: 10.1109/TPAMI.2019.2920636.

[15] Brockmann, D. Geisel, Theo. (2000). The ecology of gaze shifts. Neurocomputing. 32-33. 643-650. 10.1016/S0925-2312(00)00227-7.

[16] G. Boccignone and M. Ferraro, Modelling gaze shift as a constrained random walk, Physica A, vol. 331, no. 1, pp. 207-218, 2004

[17] Le Meur O, Liu Z. Saccadic model of eye movements for free-viewing condition. Vision Res. 2015 Nov;116(Pt B):152-64. doi: 10.1016/j.visres.2014.12.026. Epub 2015 Feb 24. PMID: 25724662

[18] Le Meur O, Coutrot A. Introducing context-dependent and spatially-variant viewing biases in saccadic models. Vision Res. 2016 Apr;121:72-84. doi: 10.1016/j.visres.2016.01.005. Epub 2016 Feb 26. PMID: 26898752

[19] The saccadic flow baseline: Accounting for image-independent biases in fixation behaviour. / Clarke, Alasdair D. F.; Stainer, Matthew J; Tatler, Benjamin W et al. In: Journal of Vision, Vol. 17, No. 11, 01.09.2017, p. 1-19.

[20] H. Liu, D. Xu, Q. Huang, W. Li, M. Xu and S. Lin, "Semantically-Based Human Scanpath Estimation with HMMs," 2013 IEEE International Conference on Computer Vision, Sydney, NSW, Australia, 2013, pp. 3232-3239, doi: 10.1109/ICCV.2013.401.

[1] https://github.com/dariozanca/FixaTons/tree/master

[2] https://github.com/matthias-k/pysaliency/tree/dev

[21] W. Sun, Z. Chen and F. Wu, ”Visual Scanpath Prediction Using IOR-ROI Recurrent Mixture Density Network,” in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 6, pp. 2101-2118, 1 June 2021, doi: 10.1109/TPAMI.2019.2956930.

[22] M. Assens, X. Giro-i-Nieto, K. McGuinness and N. E. O'Connor, ”SaltiNet: Scan-Path Prediction on 360 Degree Images Using Saliency Volumes,” 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), Venice, Italy, 2017, pp. 2331-2338, doi: 10.1109/ICCVW.2017.275.

[23] Assens, Marc et al. “PathGAN: Visual Scanpath Prediction with Generative Adversarial Networks.” ArXiv abs/1809.00567 (2018): n. pag.

[24] Kümmerer M, Bethge M, Wallis TSA. DeepGaze III: Modeling free-viewing human scanpaths with deep learning. J Vis. 2022 Apr 6;22(5):7. doi: 10.1167/jov.22.5.7. PMID: 35472130; PMCID: PMC9055565.

[25] Matthias Kümmerer, Tom Wallis, Matthias Bethge; DeepGaze II: Predicting fixations from deep features over time and tasks. Journal of Vision 2017;17(10):1147. https://doi.org/10.1167/17.10.1147.

[26] Tsotsos, John K. ; Culhane, Scan M. ; Kei Wai, Winky Yan ; Lai, Yuzhong ; Davis, Neal Nuflo, Fernando (1995). Modeling visual attention via selective tuning. Artificial Intelligence 78 (1-2):507-545.

[27] Dirk Walther and Christof Koch (2006), Modeling attention to salient proto-objects. Neural Networks 19, 1395-1407

[28] Kalesnykas RP, Sparks DL.: The Primate Superior Colliculus and the Control of Saccadic Eye Movements. The Neuroscientist. 1996;2(5):284-292. doi:10.1177/107385849600200514

[29] Kirillov, Alexander, et al. ”Segment anything.” Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023.

# VII. Supplements

## A. Saliency map implementation

Two different implementation for the Itti-Koch saliency maps are used respectively for the Itti-Koch model and CLE model, in fact, `saliency_type`, defines the type of saliency map to compute, it can be either '98' or '01'. '01' computes the saliency accordingly to the implementation provided by D. B. Walther's Saliency Toolbox derived from his PhD thesis. '98' computes the saliency accordingly to the original publication from Itti, Koch and Niebur.

## B. Installation notes

**Itti-Koch**: despite the model being available on MATLAB in the SaliencyToolBox[3] [27], the model has been re-implemented in python. The new implementation is less flexible, as the various parameters from the type of Gaussian pyramid to the number of levels are hard coded based on the values from the original Itti-Koch. Still, this implementation provides a small and easy to use class to compute saliency and predicting the scanpath.

**IOR-ROI**: weights for the network have to be downloaded[4]. Once data.zip is downloaded, extract the data folder and place it in the following directory: `pyscanpath\models\Iorroi`. Differently from the original implementation, here the segmentation process is done using META Segment Anything Model (SAM) [29], which needs to be installed separately[5]. The model checkpoint (vit_h.pth) for SAM can be downloaded from the same GitHub page[6]. Create a folder named checkpoint, place the .pth file in that folder and place the checkpoint folder in the following directory: `pyscanpath\models\Iorroi`.

[3] https://github.com/DirkBWalther/SaliencyToolbox

[4] https://mega.nz/file/KvxEXS5Z#p-ZxpjiJ6k9Tj9vxH8CGX0Ec9MQW0SJX_XSeEJcmvW0

[5] https://github.com/facebookresearch/segment-anything

[6] https://dl.fbaipublicfiles.com/segment_anything/sam_vit_h_4b8939.pth