# Deep Learning of Biomimetic Visual Perception for Virtual Humans

Masaki Nakada
University of California, Los Angeles

Honglin Chen
University of California, Los Angeles

Demetri Terzopoulos
University of California, Los Angeles

## ABSTRACT

Future generations of advanced, autonomous virtual humans will likely require artificial vision systems that more accurately model the human biological vision system. With this in mind, we propose a strongly biomimetic model of visual perception within a novel framework for human sensorimotor control. Our framework features a biomechanically simulated, musculoskeletal human model actuated by numerous skeletal muscles, with two human-like eyes whose retinas have spatially nonuniform distributions of photoreceptors not unlike biological retinas. The retinal photoreceptors capture the scene irradiance that reaches them, which is computed using ray tracing. Within the sensory subsystem of our model, which continuously operates on the photoreceptor outputs, are 10 automatically-trained, deep neural networks (DNNs). A pair of DNNs drive eye and head movements, while the other 8 DNNs extract the sensory information needed to control the arms and legs. Thus, exclusively by means of its egocentric, active visual perception, our biomechanical virtual human learns, by synthesizing its own training data, efficient, online visuomotor control of its eyes, head, and limbs to perform tasks involving the foveation and visual pursuit of target objects coupled with visually-guided reaching actions to intercept the moving targets.

## CCS CONCEPTS

• **Computing methodologies → Computer vision**; **Computer graphics**; **Animation**; **Bio-inspired approaches**; **Neural networks**; **Artificial life**; **Physical simulation**;

## KEYWORDS

Biomimetic visual perception; Computer vision; Sensorimotor control; Deep neural network learning; Biomechanical human animation.

**ACM Reference format:**
Masaki Nakada, Honglin Chen, and Demetri Terzopoulos. 2018. Deep Learning of Biomimetic Visual Perception for Virtual Humans. In *Proceedings of SAP '18: ACM Symposium on Applied Perception 2018, Vancouver, BC, Canada, August 10–11, 2018 (SAP '18),* 8 pages.
https://doi.org/10.1145/3225153.3225161

## 1 INTRODUCTION

A provocative and challenging research problem in computer animation is to enable virtual humans to perceive their photorealistic 3D virtual environments in a way similar to how we biological humans perceive our physical surroundings. Tackling this problem promises to yield autonomous virtual humans that behave more like real people. In the context of visual perception, our approach is to develop general-purpose artificial vision systems that more accurately model the human biological vision system. In this paper, we develop a strongly biomimetic model of visual perception within a novel framework for human sensorimotor control.

Biological vision has inspired computational approaches that mimic what is known about neural mechanisms underlying visual perception. Recent breakthroughs in machine learning with artificial (convolutional) neural networks have proven to be effective in computer vision; however, the application of Deep Neural Networks (DNNs) to sensorimotor systems has received little attention in either vision or graphics. Sensorimotor functionality in biological organisms refers to the continuous acquisition and interpretation of sensory information necessary to produce appropriate motor responses in order to perform actions that achieve desired goals.

Our sensorimotor framework (see Fig. 1) is unique in that it features a biomechanical human musculoskeletal model actuated by numerous skeletal muscles, with two human-like eyes whose retinas have spatially nonuniform distributions of photoreceptors. The retinal photoreceptors respond proportionally to the scene irradiance reaching them, which is computed using ray tracing. Within the sensory subsystem of our model, which continuously processes the photoreceptor outputs, are 10 automatically-trained vision DNNs that operate synergistically. A pair of DNNs, part of the oculomotor system, control eye movements, as well as head movements via the cervical muscles, while the other 8 DNNs extract the sensory information needed to control the muscles of the arms and legs.[1] Thus, driven by its egocentric, active visual perception, our biomechanical virtual human learns efficient, online visuomotor control of its eyes, head, and four limbs to perform tasks involving the foveation and visual pursuit of target objects coupled with visually-guided reaching actions to intercept the moving targets.

To our knowledge, our visuomotor control system is unprecedented not only in its ability to subserve a sophisticated biomechanical human model, but also in its use of modern machine learning methodologies to control a realistic musculoskeletal system and perform online visual processing that supports active, foveated perception, all of which is accomplished by a modular set of DNNs

*SAP '18, August 10–11, 2018, Vancouver, BC, Canada*

ACM ISBN 978-1-4503-5894-1/18/08…$15.00
https://doi.org/10.1145/3225153.3225161

[1] In the motor subsystem (bottom half of Fig. 1), two DNNs control the 216 neck muscles that balance the head atop the cervical column against the downward pull of gravity and actuate the cervicocephalic musculoskeletal complex, thereby producing controlled head movements, and 8 DNNs control the limbs; in particular, the 29 muscles in each of the two arms and the 39 muscles in each of the two legs. See our companion paper [Nakada et al. 2018] for the details.

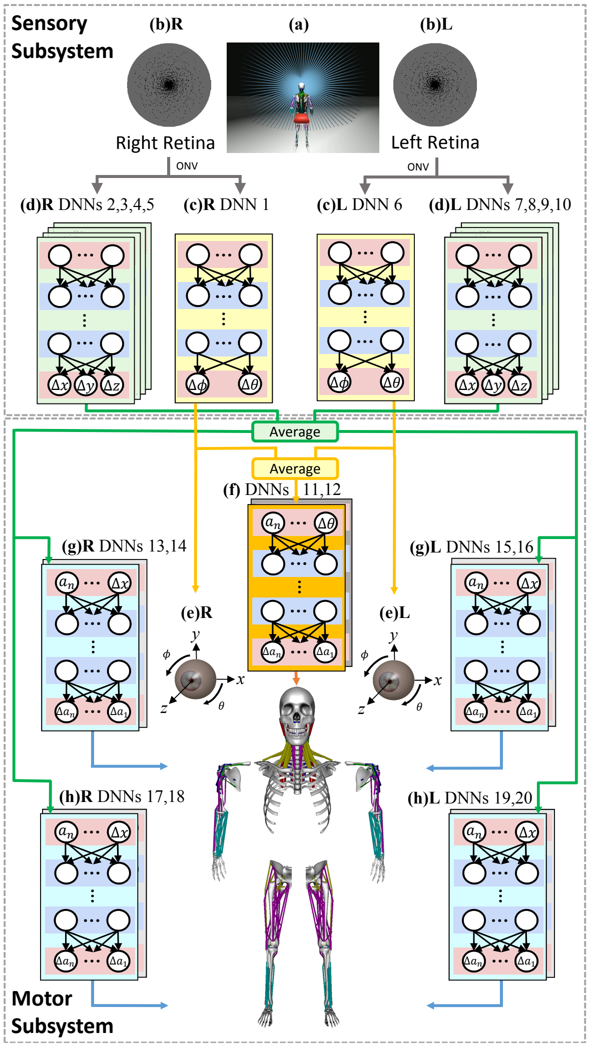


**Figure 1: The sensorimotor system architecture, whose controllers include a total of 20 DNNs, numbered 1–20.**

**• Sensory Subsystem (top): (a) Each retinal photoreceptor casts a ray into the virtual world to compute the irradiance captured by the photoreceptor. (b) The arrangement of the photoreceptors (black dots) on the left (b)L and right (b)R foveated retinas. Each retina outputs an Optic Nerve Vector (ONV). There are 10 vision DNNs. The two (yellow) foveation DNNs (c) (1,6) input the ONV and produce eye movements to foveate visual targets. (d) The other eight (green) vision DNNs—(d)L (7,8,9,10) for the left eye (e)L and (d)R (2,3,4,5) for the right eye (e)R—also input the ONV and output limb-to-target visual discrepancy estimates.**

**• Motor Subsystem (bottom): There are 10 motor DNNs. The (orange) cervicocephalic neuromuscular motor controller (f) (DNNs 11,12) inputs the average of the foveation DNN responses and outputs activations to the neck muscle group. The four (blue) limb neuromuscular motor controllers (g),(h) (DNNs 13–20) of the limb musculoskeletal complexes input the average of the left (d)L and right (d)R limb vision DNN responses and output activations to the respective arm and leg muscle groups.**

that are automatically trained from data synthesized by the human model itself.

## 2 RELATED WORK

A number of papers in the literature have explored the topic of visual perception for autonomous graphical characters. The seminal "Boids" behavioral animation model of Reynolds [1987] for animating flocks, schools, and herds, maintained group formations through perceptual awareness of the positions and velocities of nearby agents. The artificial fishes of Tu and Terzopoulos [1994] sensed their world through simulated visual perception within a limited field of view and subject to natural occlusion conditions, as did the autonomous pedestrians of Shao and Terzopoulos [2005].

Renault et al. [1990] proposed a more elaborate form of "synthetic vision" for behavioral actors, including the automatic computation of internal spatial maps of the world, and they extended their approach in subsequent efforts [Noser et al. 1995; Thalmann et al. 1997]. Among others, Kuffner and Latombe [1999], Peters and O'Sullivan [2002], Courty et al. [2003], Lozano et al. [2003], and Ondřej et al. [2010] adopted and further developed the synthetic vision approach.

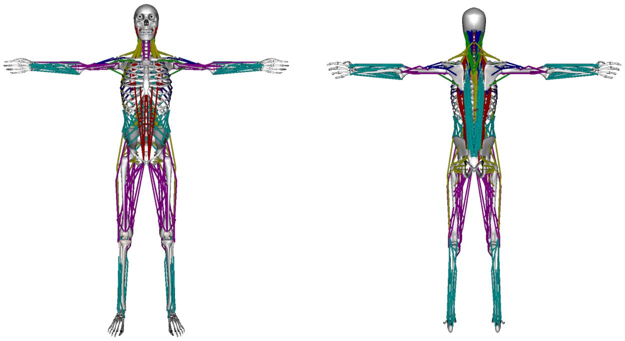
More relevant to our work is the "animat vision" approach of Terzopoulos and Rabie [1995], which employed foveated perception, eye movements, and computer vision algorithms. It was applied within a kinematic virtual human capable of bipedal locomotion, demonstrating active, vision-guided tracking and pursuit [Rabie and Terzopoulos 2000]. A similar kinematic virtual human model, named "Walter", was proposed by Sprague et al. [2007] to study visuomotor control in the context of a sidewalk navigation task.

Closely related to the work reported in the present paper, Yeo et al. [2012] developed a visuomotor system for another kinematic, anthropomorphic virtual character, which was capable of visual target estimation tasks and demonstrated realistic ball catching actions; however, the character predicts the trajectories of thrown balls from their known positions and velocities in 3D space, without performing any biologically-inspired visual processing.

By contrast, our work is the first to employ a fully dynamic (as opposed to purely kinematic), biomechanically-simulated, human musculoskeletal model. This presents a much more biomimetic and difficult visuomotor control problem, especially so for human-like eyes, capable of eye movements, that are part of a cervicocephalic complex actuated by 216 muscles. Moreover, we are the first to attempt a deep learning approach to tackling this problem.

Unlike the uniform Cartesian grid visual sampling of synthetic vision techniques and artificial imaging sensors, visual sampling in the primate retina is known to be strongly space-variant [Schwartz 1977]. The density of cone photoreceptors decreases radially from the fovea toward the periphery. Log-polar photoreceptor distributions are a common model of space-variant image sampling [Grady 2004; Koenderink and Van Doorn 1978; Wilson 1983].

The virtual humans demonstrated by Rabie and Terzopoulos [2000] were equipped with eyes implemented as coaxial virtual cameras that rendered polygon-shaded images via the GPU pipeline, thus yielding multiresolution pyramids supporting foveal/peripheral vision. Our retinal model is significantly more biomimetic. Given their fundamentally nonuniform distributions of photoreceptors,



**(a) Frontal view** **(b) Dorsal view**

**Figure 2: The biomechanical model, showing the musculoskeletal system with its 193 bones and 823 Hill-type muscle actuators.**

specifically noisy log-polar distributions, the retinas in our eye models sample the 3D scene using ray tracing, which better emulates how the human retina samples scene radiance from the incidence of light on its photoreceptors. Ray tracing has also been applied in ophthalmology as a methodology for synthesizing retinal images in order to predict changes in visual performance due to changes in the eyes [Greivenkamp et al. 1995; Wei et al. 2014].

## 3 BIOMECHANICAL MUSCULOSKELETAL HUMAN MODEL

Fig. 2 shows the anatomically accurate musculoskeletal system of our human model. It includes all of the relevant articular bones and muscles—193 bones connected by joints comprising 163 articular degrees of freedom, plus a total of 823 muscle actuators. Each skeletal muscle is modeled as a Hill-type uniaxial contractile actuator that applies forces to the bones at its points of insertion and attachment.[2] The human model is numerically simulated as a force-driven articulated multi-body system (refer to [Lee et al. 2009] for the details).

Each muscle actuator is activated by an independent, time-varying, efferent activation signal $a(t)$. Given our human model, the overall challenge in neuromuscular motor control is to determine the activation signals for each of its 823 muscles necessary to carry out various motor tasks. For now, we mitigate complexity by placing our virtual human in a seated position, immobilizing the pelvis as well as the lumbar and thoracic spinal column vertebra and other bones of the torso, leaving the cervicocephalic, arm, and leg neuromuscular complexes free to articulate.

Additional details about our biomechanical human musculoskeletal model and the 5 neuromuscular motor controllers comprising its motor subsystem (see Footnote 1 and the lower half of Fig. 1) are presented in our companion paper [Nakada et al. 2018]. The remainder of the present paper develops in greater detail the sensory subsystem, which is illustrated in the top half of Fig. 1. Currently, the sensory subsystem is purely visual.

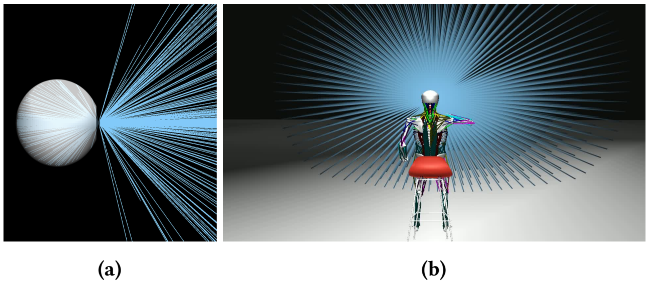


**Figure 3: (a) Rays cast from the positions of photoreceptors on the retina through the pinhole aperture and out into the scene by the ray tracing procedure that computes the irradiance responses of the photoreceptors. (b) Rays cast from both eyes as the seated virtual human looks forward.**

## 4 EYE AND RETINA MODELS

*Eye model:* We modeled the eyes in accordance with human physiological data.[3] As shown in Fig. 1e, we model the virtual eye as a sphere of 12mm radius that can be rotated with respect to its center around its vertical $y$ axis by a horizontal angle of $\theta$ and around its horizontal $x$ axis by a vertical angle of $\phi$. The eyes are in their neutral positions, looking straight ahead, when $\theta = \phi = 0°$. We currently model the eye as an ideal pinhole camera with aperture (optical center) at the center of the pupil and with horizontal and vertical fields of view of 167.5°.

We compute the irradiance at any point on the spherical retinal surface at the back of the eye using conventional ray tracing. Sample rays from the positions of photoreceptors on the retinal surface are cast through the pinhole and out into the 3D virtual world, where they recursively intersect with the visible surfaces of virtual objects and, in accordance with the Phong local illumination model, combine with shadow rays to light sources. The RGB values returned by these rays determine the irradiance impinging upon the retinal photoreceptors. Fig. 3 illustrates the retinal "imaging" process.

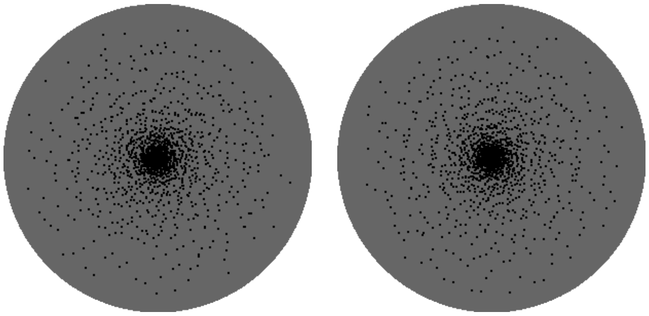
*Placement of the Photoreceptors:* To emulate biomimetic foveated vision, we procedurally position the photoreceptors on the hemispherical retina according to a noisy log-polar distribution, which has greater biological fidelity compared to earlier foveated vision models [Rabie and Terzopoulos 2000]. On each retina, we include 3,600 photoreceptors situated at

$$\boldsymbol{d}_k = e^{\rho_j} \begin{bmatrix} \cos\alpha_i \\ \sin\alpha_i \end{bmatrix} + \begin{bmatrix} \mathcal{N}(\mu, \sigma^2) \\ \mathcal{N}(\mu, \sigma^2) \end{bmatrix}, \quad \text{for } 1 \leq k \leq 3{,}600, \qquad (1)$$

where $0 < \rho_j \leq 40$, incremented in steps of 1, and $0 \leq \alpha_i < 360°$, incremented in 4° steps, and where $\mathcal{N}$ denotes additive IID Gaussian noise. We set mean $\mu = 0$ and variance $\sigma^2 = 0.0025$, which places the photoreceptors in slightly different positions on the two retinas.

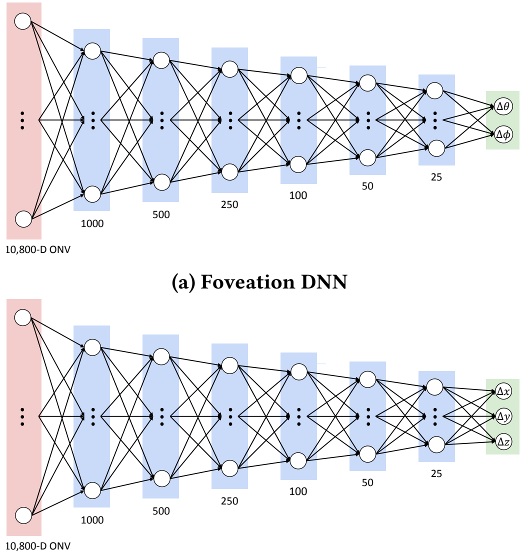
[2] The muscle actuators are embedded in a finite element model of the musculotendinous soft tissues of the body that produces realistic flesh deformations. For the purposes of the research reported in the present paper, however, the finite element soft-tissue simulation is unnecessary and it is excluded in order to reduce the computational cost.

[3] The transverse size of an average eye is 24.2 mm and its sagittal size is 23.7 mm. The approximate field of view of an individual eye is 100 degrees to temporal, 45 degrees to nasal, 30 degrees to superior, and 70 degrees to inferior. The combined field of view of the two eyes is approximately 200 degrees horizontally and 135 degrees vertically.



(a) Left retina (b) Right retina

Figure 4: Positions of the photoreceptors (black dots) on the retinas according to the noisy log-polar model.



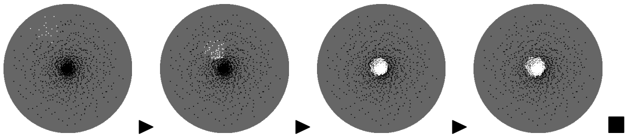
(a) Foveation DNN

(b) Limb vision DNN

Figure 5: The vision DNN architecture.

Fig. 4 illustrates the placement of the photoreceptors on the left and right retinas. Other placement patterns are readily implementable, including more elaborate procedural models [Deering 2005] or photoreceptor distributions empirically measured from biological eyes, all of which differ dramatically from the uniformly-sampled rectangular images common in computer graphics and vision.

*Optic nerve vectors:* The foveated retinal RGB "image" captured by each eye is output for further processing down the visual pathway, not as a 2D array of pixels, but as a 1D vector of length $3{,}600 \times 3 = 10{,}800$, which we call the Optic Nerve Vector (ONV). The raw sensory information encoded in the ONV feeds the vision DNNs that directly control eye movements and feed the neuromuscular motor controller networks that orchestrate neck-actuated head motions and the actions of the limbs.



(a) $t_0$ (b) $t_1$ (c) $t_2$ (d) $t_3$

Figure 6: Time sequence (a)–(d) of photoreceptor responses in the left retina during a saccadic eye movement that foveates and tracks a moving white sphere. At time $t_0$ the sphere becomes visible in the periphery, at $t_1$ the eye movement is bringing the sphere toward the fovea, and the moving sphere is being fixated in the fovea at times $t_2$ and $t_3$.

## 5 VISION DNNs (1–10)[4]

Fig. 1 overviews the sensorimotor system, which comprises sensory and motor subsystems, and its caption describes the information flow and the functions of the system's 20 DNN controllers (numbered 1–20 in the figure). The previous section presented the details of the eyes (Fig. 1e) and their retinas (Fig. 1b). Next, we will discuss in greater detail the 10 vision DNNs (numbered 1–10 in Fig. 1).

The sensory subsystem includes two types of vision DNNs. Both input the visual information provided by the 10,800-dimensional ONV. Through a systematic set of experiments (see Section 7), we identified a common DNN architecture (Fig. 5) that works well for all 10 vision DNNs—tapered, feedforward, fully-connected networks with six hidden layers of rectified linear units (ReLUs).

The first type of vision DNNs are foveation DNNs that control eye movements, as well as head movements via the cervicocephalic neuromuscular motor controller. The second type are limb vision DNNs, which produce arm-to-target 3D discrepancies, $\Delta x$, $\Delta y$, and $\Delta z$, that drive limb actions via the limb neuromuscular motor controllers. Both types are described in the next two sections.

### 5.1 Foveation DNNs (1,6)

Along with the eyes, the left and right foveation DNNs constitute the oculomotor subsystem. The first role of these DNNs is to produce voluntary changes in gaze direction by driving saccadic eye movements to foveate visible objects of interest, thereby observing them with maximum visual acuity. This is illustrated in Fig. 6 for a white sphere in motion that enters the field of view from the lower right, stimulating some peripheral photoreceptors at the upper left of the retina. The eye almost instantly performs a saccadic rotation to foveate the visual target. Fine adjustments comparable to microsaccades are observed during fixation.

To aid foveation, fixation, and visual tracking, eye movements induce compensatory head movements, albeit much more sluggish ones due to the considerable mass of the head. Hence, the second role of the foveation DNNs is to control head movements, by driving the cervicocephalic neuromuscular voluntary motor DNN (11) (Fig. 1f) with the average of their two outputs.

As shown in Fig. 5a, the input layer to this DNN has 10,800 units in accordance with the dimensionality of the ONV, the output layer has 2 units representing eye rotation adjustments, $\Delta\theta$ and $\Delta\phi$, and there are 6 hidden layers with unit counts as indicated in the figure.

[4] The numbers in parentheses here and elsewhere refer to numbered DNNs in Fig. 1.

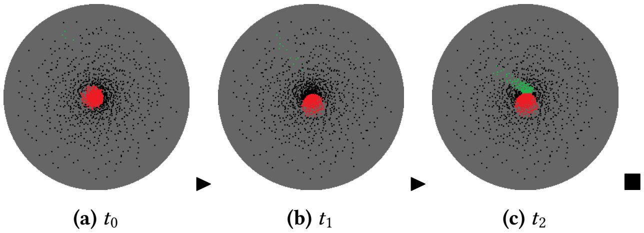


**Figure 7: Photoreceptor responses during an arm-reaching motion toward a moving target sphere. The photoreceptors are simultaneously stimulated by the fixated red sphere and by the green arm entering the eye’s field of view from the lower right (upper left on the retina).**

We chose this network architecture as a result of a systematic set of experiments reported in Section 7.

We use our human model to train the network, as follows: We present a white sphere within the visual field. The responses of the photoreceptors in the retinas of each eye are computed by ray tracing the 3D scene, and they are output as the RGB components of the respective ONV. Given its ONV input, the desired output of the network is the angular discrepancies, $\Delta\theta$ and $\Delta\phi$, between the actual gaze directions of the eyes and the known gaze directions that would foveate the sphere. Repeatedly positioning the sphere at random locations in the visual field, we generated a large training dataset of 1M input-output pairs. The backpropagation DNN training process (see Footnote 5) converged to a small error after 80 epochs before triggering the early stopping condition to avoid overfitting.

## 5.2 Limb Vision DNNs (2,3,4,5 & 7,8,9,10)

The role of the left and right limb (arm and leg) vision DNNs is to estimate the separation in 3D space between the position of the end effector (hand or foot) and the position of a visual target, thus driving the associated limb neuromuscular motor controller to extend the limb to touch the target. This is illustrated in Fig. 7 for a fixated red sphere and a green arm that enters the eye’s field of view from the lower right, stimulating peripheral photoreceptors at the upper left of the retina.

The architecture of the limb vision DNNs, shown in Fig. 5b, is identical to the foveation DNNs, except for the size of the output layer, which has 3 units, $\Delta x$, $\Delta y$, and $\Delta z$, the estimated discrepancies between the 3D positions of the end effector and visual target.

Again, we use our biomechanical human musculoskeletal model to train the four limb vision DNNs, as follows: A red sphere is presented in the visual field and the trained foveation DNNs are allowed to foveate the sphere. Then, a limb (arm or leg) is extended toward the sphere. Again, the responses of the photoreceptors in the retinas of the eyes are computed by ray tracing the 3D scene, and they are output as the RGB components of the respective ONV. Given the ONV input, the desired output of the network is the discrepancies, $\Delta x$, $\Delta y$, and $\Delta z$, between the known 3D positions of the end effector and visual target. Repeatedly placing the sphere at random positions in the visual field and randomly articulating the limb to reach for it in space, we generated a training dataset of 1M input-output pairs. The backpropagation DNN training process converged to a small error after 388 epochs, which triggered the early stopping condition to avoid overfitting. As expected, due to the higher complexity of this task, the training is significantly slower than for the foveation DNN.

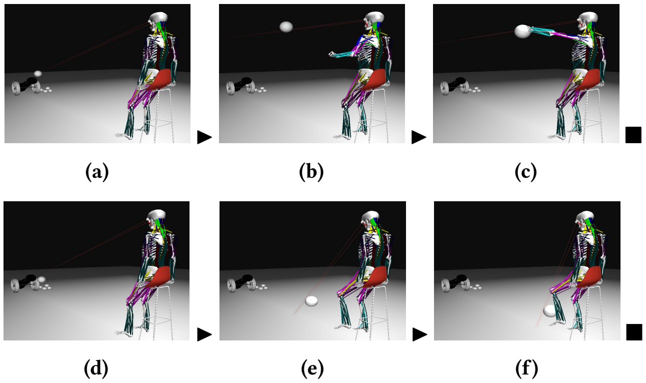
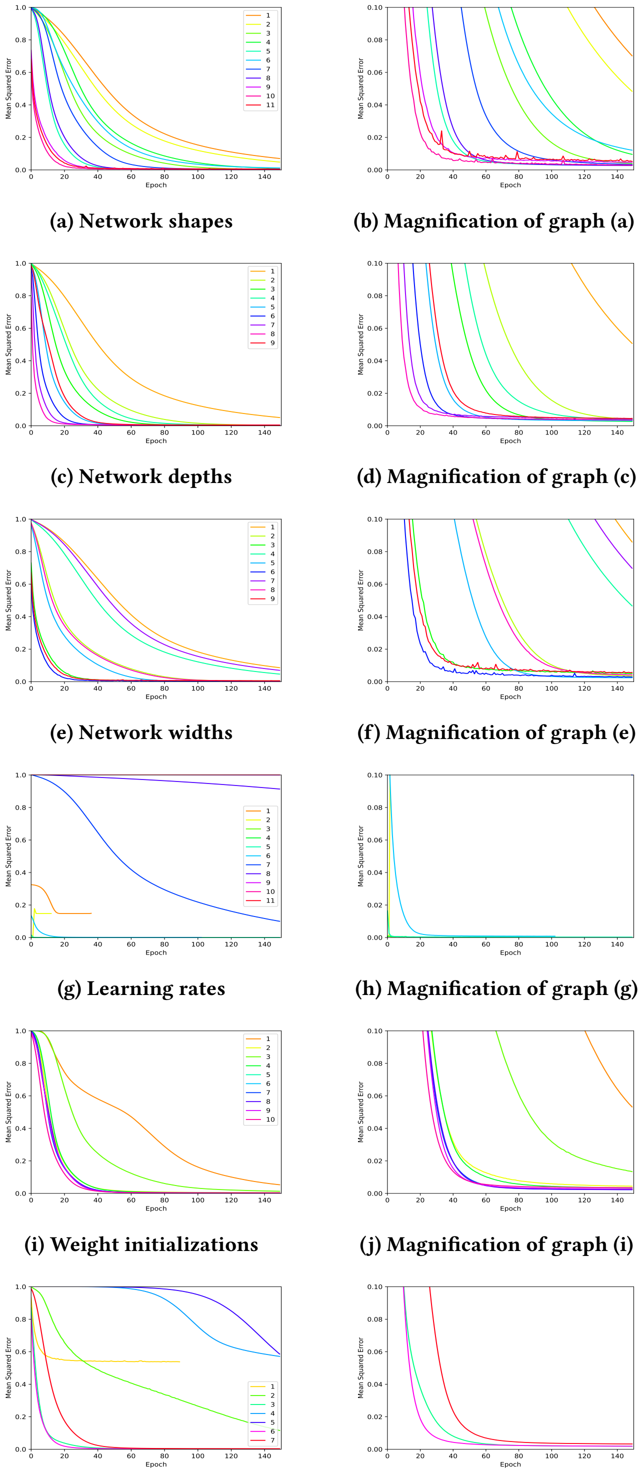


**Figure 8: Sequences of frames rendered from a simulation of the biomechanical virtual human sitting on a stool, demonstrating active visual sensory and motor responses—a left-arm reaching action (a)–(c) and a left-leg kicking action (d)–(f) to intercept balls shot at it by a cannon. The balls are observed by the eyes via the vision DNNs, foveated and tracked through eye movements in conjunction with muscle-actuated head movements controlled by the cervicocephalic neuromuscular motor controller, while visually guided, muscle-actuated limb movements are controlled by the left arm and left leg neuromuscular motor controllers. The muscles are color coded to distinguish the different muscle groups, with brightness proportional to each contractile muscle’s efferent neural activation.**

# 6 RESULTS

Fig. 8 shows a sequence of frames from a simulation demonstrating the functionality of the sensorimotor system. A cannon shoots balls at the virtual human, which it actively perceives with its eyes and reaches out with its arms and legs to intercept. Its 20 DNNs operate continuously and synergistically. The ONVs from the retinas are processed by the pair of foveation DNNs, driving the foveation and visual pursuit of the incoming balls through eye movements coupled with cooperative head movements that pursue the gaze direction. The head movements are controlled by the cervicocephalic neuromuscular motor controller, which is fed by the average of the foveation DNN outputs. Naturally, the head movements are much more sluggish than the eye movements due to the considerable mass of the head. Simultaneously, guided by the outputs of the four pairs of limb vision DNNs, the neuromuscular limb motor controllers actuate the arms and legs such that they extend to intercept the incoming balls, deflecting them out of the way. Thus, the biomechanical human musculoskeletal model continuously controls itself, in gravity, to carry out this nontrivial sensorimotor task in an online, (virtual) real-time manner, and no balls shot at it are missed.

Additional experiments and demonstrations are presented in our companion paper [Nakada et al. 2018].



(a) Network shapes

(b) Magnification of graph (a)

(c) Network depths

(d) Magnification of graph (c)

(e) Network widths

(f) Magnification of graph (e)

(g) Learning rates

(h) Magnification of graph (g)

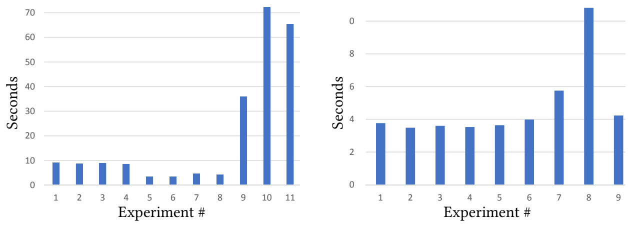
(i) Weight initializations

(j) Magnification of graph (i)

(k) Activation functions

(l) Magnification of graph (k)

**Figure 9: Training progress of the foveation DNN in experiments with various network shapes (a),(b); depths (c),(d); widths (e),(f); learning rates (g),(h); weight initializations (i),(j); and activation functions (k),(l). Graphs on the right magnify the bottom 10% of the graphs on the left.**



**(a)** **(b)**

**Figure 10: Average time per epoch (a) for networks of various shapes and (b) for networks of various depths.**

## 7 VISION DNN EXPERIMENTS

We conducted a systematic set of experiments to compare the performance of various possible vision DNN architectures and training techniques. The networks were implemented and trained using the Theano library [Bergstra et al. 2010] running on an NVIDIA Titan X GPU installed in a Ubuntu 16.04 system.[5]

This section presents our experiments with the foveation DNNs. For each experiment, we randomly selected a 0.1M input-output-pair validation subset from the 1M input-output-pair synthesized training dataset. The smaller datasets that we employed reduced the computational load, thus enabling more rapid experimentation.

*Network shape:* We conducted training experiments with the Foveation DNN controller using 11 different network architectures of various shapes. The network shapes are as follows, where the leftmost number, rightmost number, and intermediate numbers indicate the number of units in the input layer, output layer, and hidden layers, respectively:

(1) 10,800 | 100 | 200 | 2
(2) 10,800 | 200 | 100 | 2
(3) 10,800 | 300 | 200 | 100 | 2
(4) 10,800 | 100 | 200 | 300 | 2
(5) 10,800 | 500 | 250 | 100 | 50 | 25 | 2
(6) 10,800 | 25 | 50 | 100 | 250 | 500 | 2
(7) 10,800 | 100 | 250 | 500 | 250 | 100 | 2
(8) 10,800 | 500 | 250 | 100 | 250 | 500 | 2
(9) 10,800 | 10,000 | 10,000 | 2
(10) 10,800 | 20,000 | 10,000 | 2
(11) 10,800 | 10,000 | 20,000 | 2

The list encompasses deep and shallow as well as narrow and wide network architectures with straight, regular, inverse triangle, and diamond shapes. Fig. 9a graphs the mean squared error as a function of the number of epochs during the training process on the validation datasets for each of the above listed network architectures, by number. All the training processes converged; however, the convergence speed and stability varied across the architectures. The shallow-narrow neural networks (1,2,3,4) required the largest number of epochs to converge. The progress of learning was stable but slow. The shallow-wide neural networks (9,10,11) where

[5] The DNNs employ rectified linear units (i.e., the ReLU activation function). The total number of weights in the vision DNNs is 11,458,227 for the foveation DNNs (Fig. 5a) and 11,458,253 for the limb vision DNNs (Fig. 5b). The initial weights are sampled from the zero-mean normal distribution with standard deviation $\sqrt{2/fan\_in}$, where $fan\_in$ is the number of input units in the weight tensor [He et al. 2015]. To train the DNNs, we apply the mean-squared-error loss function and the Adaptive Moment Estimation (Adam) stochastic optimizer [Kingma and Ba 2014] with learning rate $\eta = 10^{-6}$, step size $\alpha = 10^{-3}$, forgetting factors $\beta_1 = 0.9$ for the gradients and $\beta_2 = 0.999$ for their second moments, and avoid overfitting using the early stopping condition of negligible improvement for 10 successive epochs. Each epoch requires less than 10 seconds of computation time.

more efficient but less stable, in the sense that they required fewer epochs, but the error did not decrease monotonically, as we see in Fig. 9b. The deep architectures (5,6,7,8) converged quickly and stably. The deep tapered network (5) converged with the best speed and stability. The average time spent for each epoch is shown in Fig. 10a. As expected, it increases with the number of weights, wide networks requiring 10 to 20 times more computation time per epoch than narrow ones. However, using a trained wide network for online control allows parallelization, since the outputs of units in the same layer can be computed independently, although the layers must be computed sequentially, rendering deeper networks less parallelizable. Overall, the training times for the more elaborate architectures were similar to those for the simpler ones, since the former required fewer epochs, albeit more time per epoch, whereas the latter required more epochs, but less time per epoch.

*Network depth:* The next set of experiments evaluated networks of the same tapered shape, but different depths, as follows:

(1) 10,800 | 200 | 100 | 2
(2) 10,800 | 300 | 200 | 100 | 2
(3) 10,800 | 400 | 300 | 200 | 100 | 2
(4) 10,800 | 500 | 250 | 100 | 50 | 25 | 2
(5) 10,800 | 1,000 | 500 | 250 | 100 | 50 | 25 | 2
(6) 10,800 | 1,800 | 1,000 | 500 | 250 | 100 | 50 | 25 | 2
(7) 10,800 | 3,000 | 1,800 | 1,000 | 500 | 250 | 100 | 50 | 25 | 2
(8) 10,800 | 4,500 | 3,000 | 1,800 | 1,000 | 500 | 250 | 100 | 50 | 25 | 2
(9) 10,800 | 680 | 550 | 430 | 320 | 230 | 150 | 80 | 40 | 20 | 10 | 2

As necessary, we included larger layers after the input layer so as to increase the number of layers while maintaining the tapered shape. Fig. 9c graphs the mean squared error as a function of the number of epochs during the training process. Interestingly, fewer epochs were necessary with increasing network depth. In Experiment 9, we decreased the number of units in each layer by an order of magnitude and observed that the number of epochs required increased significantly. These results show that both the number of layers and the number of units per layer—that is, the total number of weights—contribute to the regression abilities of the vision network. The average compute time spent on each epoch is shown in Fig. 10b. This was less than 3.7 seconds for each epoch, until the number of hidden layers is increased to 7. Networks with more than 7 hidden layers required significantly more time per epoch. This is due to the complexity of the architecture and the larger number of weights to train. For Experiment 9, with the tenfold decrease in the number of units, the training time for each epoch again decreases to around 4 seconds.

Hence, to strike a good compromise between accuracy, efficiency, and stability, we decided to use a 6-hidden-layer network architecture.

*Network width:* The next set of experiments evaluated networks of the same depth, but different widths, as follows:

(1) 10,800 | 100 | 100 | 2
(2) 10,800 | 1,000 | 1,000 | 2
(3) 10,800 | 10,000 | 10,000 | 2
(4) 10,800 | 200 | 100 | 2
(5) 10,800 | 2,000 | 1,000 | 2
(6) 10,800 | 20,000 | 10,000 | 2
(7) 10,800 | 100 | 200 | 2
(8) 10,800 | 1,000 | 2,000 | 2
(9) 10,800 | 10,000 | 20,000 | 2

Fig. 9e graphs the network training convergences. We see that the wider the architectures, the fewer epochs are required. This is as expected, because the network's representation capability should increase as the number of nodes in each layer increases. The results show that the shape of wide neural networks (straight, tapered, or inverse tapered) has only a minor influence. Although most of these networks converged well, the training progress was not as efficient and stable as for the deeper architectures.

*Learning rate:* We conducted experimental trainings with the foveation DNN using 11 different architectures to determine the best learning rate with Adam optimization. The number of hidden layers was fixed to 6 with the tapered DNN used in the network depth experiment. Only the learning rates for the stochastic optimization differ. We choose 11 different learning rates decreasing by factors of 10, such that, for $1 \leq n \leq 11$, Experiment $n$ employs a learning rate of $10^{-n}$. Fig. 9g graphs the network training convergences. It can be seen that the training is most efficient and stable when the learning rate is $10^{-6}$. For learning rates less than $10^{-8}$, the trainings do not converge. For learning rates greater than $10^{-4}$, the trainings converged very quickly; however, they appear unstable and may not work robustly for other datasets.

Hence, to achieve a good balance between stability and efficiency, we chose $10^{-6}$ as the Adam learning rate in our offline training process.

*Weight initialization:* The next set of experiments evaluated the following different weight initialization methods using networks with the tapered-shape, 6-hidden-layer architecture used in the network depth experiment:

(1) Uniform
(2) LeCun uniform
(3) Normal
(4) Orthogonal
(5) Zeros
(6) Ones
(7) He Uniform
(8) He Normal
(9) Glorot uniform
(10) Glorot normal

Fig. 9i graphs the network training convergences. Although most of the training processes worked well, those with Uniform and Normal weight initialization were slower to converge. The Zeros initialization method does not converge until near the end.

Hence, we decided to employ He Normal initialization, because it yields fast and stable convergence and, indeed, it is one of the most popular weight initialization methods.

*Activation function:* The next set of experiments evaluated the following different activation functions using networks with the tapered-shape, 6-hidden-layer architecture used in the network depth experiments:

(1) Linear
(2) Soft Plus
(3) Soft Sign
(4) Hard Sigmoid
(5) Sigmoid
(6) Tanh
(7) ReLU

Fig. 9k graphs the training convergences. Networks using ReLU, Soft Sign, and Tanh activation functions show fast convergence. Those using Linear, Sigmoid, and Hard Sigmoid do not converge. The network using Soft Plus converged, but slowly.

Although Tanh and Soft Sign required fewer epochs in this experiment, we decided to use ReLU activation for its known advantages of sparsity of the representation, absence of vanishing gradients, greater biological plausibility, and computational simplicity, which leads to superior stability and robustness.

# 8 CONCLUSION

We have presented a simulation framework for investigating biomimetic human sensory and sensorimotor control. Our framework is unique in that it features an anatomically accurate, biomechanically simulated, human musculoskeletal model that is actuated by numerous contractile skeletal muscles. Our contributions in this paper include the following primary ones:

- The development of a biomimetic, foveated retina model, which is deployed in a pair of human-like eyes capable of realistic eye movements, that employs ray tracing to compute the irradiance captured by a multitude of nonuniformly arranged photoreceptors.
- Demonstration of the performance of our sensorimotor system in tasks that simultaneously involve eye movement control for saccadic foveation and pursuit of visual targets in conjunction with cooperative, dynamic head motion control, plus visually-guided dynamic limb control to produce natural arm and leg extension actions that enable the virtual human to intercept moving target objects.

## 8.1 Future Work

Our current eye model is an ideal pinhole camera. We plan to create a more realistic eye model that not only has a finite-aperture pupil that dilates and constricts to accommodate to the incoming light intensity, but also includes cornea and lens components to refract light rays and is capable of adjusting the optical depth of field through active, ciliary muscle control of the lens deformation. Furthermore, our current eye model is a purely kinematic rotating sphere. We plan to implement a biomechanical eye model with the typical 7.5 gram mass of a human eyeball, actuated by extraocular muscles, not just the 4 rectus muscles to induce most of the $\theta$, $\phi$ movement of our current kinematic eyeball, but also the 2 oblique muscles to induce torsion movements around the gaze direction.

Our biomimetic vision system generates saccadic eye movements to foveate objects of interest in a variety of different scenarios. Hence, our model can be valuable in human visual attention research, a topic that we wish to explore in future work. In this context, of most relevance to our approach would be models of visual attention that are based on deep learning, ideally through irregular convolutional neural networks that conform to nonuniformly distributed retinal photoreceptors.

The tasks of the DNNs, which must estimate from their ONV inputs the discrepancy between the 3D positions of the end effector and visual target, are made difficult by the fact that 3D depth information is naturally lost with projection onto the 2D retina and, in fact, the estimation of depth discrepancy is currently rather poor. This limitation provides an opportunity to explore binocular stereopsis with an enhanced version of our foveated perception model. For this, as well as for other types of subsequent visual processing, we will likely want to increase the number of photoreceptors, experiment with different nonuniform photoreceptor organizations, and automatically construct 2D retinotopic maps from the 1D ONV inputs.

# ACKNOWLEDGMENTS

We are indebted to Sung-Hee Lee for creating the biomechanical human musculoskeletal model that we have employed in our work. We thank Gestina Yassa for her voice over of our demonstration video, and Tao Zhou and Tomer Weiss for discussions. We gratefully acknowledge financial support from VoxelCloud, Inc., and a 5-year Scholarship for Long-Term Study Abroad to MN from the Japan Student Organization (JASSO).

# REFERENCES

J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio. 2010. Theano: A CPU and GPU math compiler in Python. In *Proc. 9th Python in Science Conference*. Austin, TX, 1–7.

N. Courty, E. Marchand, and B. Arnaldi. 2003. A new application for saliency maps: Synthetic vision of autonomous actors. In *Proc. IEEE Inter. Conf. Image Processing*, Vol. 3. Barcelona, Spain, 1065.

M. F. Deering. 2005. A photon accurate model of the human eye. *ACM Transactions on Graphics* 24, 3 (2005), 649–658.

L.J. Grady. 2004. *Space-variant computer vision: A graph-theoretic approach*. Ph.D. Dissertation. Boston University.

J.E. Greivenkamp, J. Schwiegerling, J.M. Miller, and M.D. Mellinger. 1995. Visual acuity modeling using optical raytracing of schematic eyes. *American Journal of Ophthalmology* 120, 2 (1995), 227–240.

K. He, X. Zhang, S. Ren, and J. Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proc. IEEE International Conference on Computer Vision*. Santiago, Chile, 1026–1034.

D. Kingma and J. Ba. 2014. *Adam: A method for stochastic optimization*. Technical Report. arXiv preprint arXiv:1412.6980.

J.J. Koenderink and A.J. Van Doorn. 1978. Visual detection of spatial contrast; influence of location in the visual field, target extent and illuminance level. *Biological Cybernetics* 30, 3 (1978), 157–167.

J.J. Kuffner and J.-C. Latombe. 1999. Fast synthetic vision, memory, and learning models for virtual humans. In *Proc. IEEE/CGS Inter. Conf. Computer Animation*. Geneva, Switzerland, 118–127.

S.-H. Lee, E. Sifakis, and D. Terzopoulos. 2009. Comprehensive biomechanical modeling and simulation of the upper body. *ACM Trans. on Graphics* 28, 4 (2009), 99:1–17.

M. Lozano, R. Lucia, F. Barber, F. Grimaldo, A. Lucas, and A. Fornes. 2003. An efficient synthetic vision system for 3D multi-character systems. In *Intelligent Virtual Agents (IVA 2003)*. Lecture Notes in Computer Science, Vol. 2792. Springer, Berlin, 356–357.

M. Nakada, T. Zhou, H. Chen, T. Weiss, and D. Terzopoulos. 2018. Deep learning of biomimetic sensorimotor control for biomechanical human animation. *ACM Transactions on Graphics* 37, 4, Article 56 (August 2018), 15 pages. Proc. *ACM SIGGRAPH 2018*, Vancouver, BC, August 2018.

H. Noser, O. Renault, D. Thalmann, and N.M. Thalmann. 1995. Navigation for digital actors based on synthetic vision, memory, and learning. *Computers & Graphics* 19, 1 (1995), 7–19.

J. Ondřej, J. Pettré, A.-H. Olivier, and S. Donikian. 2010. A synthetic-vision based steering approach for crowd simulation. *ACM Trans. on Graphics* 29, 4 (2010), 123.

C. Peters and C. O'Sullivan. 2002. Synthetic vision and memory for autonomous virtual humans. *Computer Graphics Forum* 21, 4 (2002), 743–752.

T.F. Rabie and D. Terzopoulos. 2000. Active perception in virtual humans. In *Proc. Vision Interface 2000*. Montreal, Canada, 16–22.

O. Renault, N.M. Thalmann, and D. Thalmann. 1990. A vision-based approach to behavioural animation. *Computer Animation and Virtual Worlds* 1, 1 (1990), 18–21.

C.W. Reynolds. 1987. Flocks, herds and schools: A distributed behavioral model. *Computer Graphics* 21, 4 (1987), 25–34. Proc. ACM SIGGRAPH '87.

E.L. Schwartz. 1977. Spatial mapping in the primate sensory projection: Analytic structure and relevance to perception. *Biological Cybernetics* 25, 4 (1977), 181–194.

W. Shao and D. Terzopoulos. 2005. Autonomous pedestrians. In *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Los Angeles, CA, 19–28.

N. Sprague, D. Ballard, and A. Robinson. 2007. Modeling embodied visual behaviors. *ACM Transactions on Applied Perception (TAP)* 4, 2 (2007), 11.

D. Terzopoulos and T.F. Rabie. 1995. Animat vision: Active vision with artificial animals. In *Proc. 5th Inter. Conf. on Computer Vision (ICCV'95)*. Cambridge, MA, 840–845.

D. Thalmann, H. Noser, and Z. Huang. 1997. Autonomous virtual actors based on virtual sensors. In *Creating Personalities for Synthetic Actors*. Springer, Berlin, 25–42.

X. Tu and D. Terzopoulos. 1994. Artificial fishes: Physics, locomotion, perception, behavior. In *Proc. ACM SIGGRAPH 94 Conf.* Orlando, FL, 43–50.

Q. Wei, S. Patkar, and D.K. Pai. 2014. Fast ray-tracing of human eye optics on graphics processing units. *Comp. Meth. and Prog. in Biomedicine* 114, 3 (2014), 302–314.

S.W. Wilson. 1983. On the retino-cortical mapping. *International Journal of Man-Machine Studies* 18, 4 (1983), 361–389.

S.H. Yeo, M. Lesmana, D.R. Neog, and D.K. Pai. 2012. Eyecatch: simulating visuomotor coordination for object interception. *ACM Trans. on Graphics* 31, 4 (2012), 42.