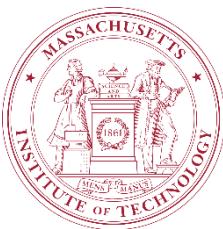


Generative Modeling

Vision, Learning, and Statistics for AI — Research Notes

Naveen Goela

negoela@alum.mit.edu



Part I

Generative Modeling and AI

AI Systems based on Information-Theoretic Principles

Digital Codecs, Encoder-Decoder Systems	  HARVARD MEDICAL SCHOOL	5G Qualcomm snapdragon	Academia, R&D
Generative Models, Variational Lower Bounds	AISTATS-2019 		Academia, R&D
Information Bottleneck Mutual Information	Research Notes		Academia, R&D

High-Level Research Proposal – Naveen Goela

Research Direction (R&D):

Generative modeling using information-theoretic principles and tools. Decompose and analyze models using information theory. **Apply variational bounds, information-theoretic training (de-noising diffusion models, VAE).**

Encoder-Decoder System in Communications

N. Goela, Qualcomm ImpaQt Innovation Award, 2014.

N. Goela, E. Abbe, M. Gastpar, **Polar Codes for Broadcast Channels**, *IEEE Transactions On Information Theory*, vol. 61, no. 2, pp. 758-782, 2015.

N. Goela, M. Raginsky, **Channel Polarization Through The Lens Of Blackwell Measures**, *IEEE Transactions On Information Theory*, vol. 66, no. 10, pp. 6222-6241, 2020.

US-9722651-B2: N. Goela, V. Ekambaram, R. Tandra, J. Soriaga, **An Adaptive Channel Coding System Using Polarization** - Lead Inventor, Qualcomm 5G Technology, 2017.

Encoder-Decoder System in DNA

N. Goela, J. Bolot, **Technicolor Visionary Award**, 2016-2017.

H. H. Lee*, R. Kalhor*, N. Goela*, J. Bolot, G. Church, **Terminator-Free Template-Independent Enzymatic DNA Synthesis for Digital Information Storage**, *Nature Communications*, (10), 2383, 2019.
* Equal contribution by authors.

WO-2018-148260-A1: N. Goela, J. Bolot, **Apparatus, Method, and System for Digital Information Storage in DNA** - Lead Inventor, Digital codec technology for DNA molecular storage, 2018.

Encoder-Decoder System in Signal Processing

N. Goela, M. Gastpar, **Reduced Dimension Linear Transform Coding of Correlated Signals in Networks**, *IEEE Transactions on Signal Processing*, vol. 60, no. 6, pp. 3174-3187, June, 2012.

Encoding and Decoding Systems in Communications

34

The Mathematical Theory of Communication

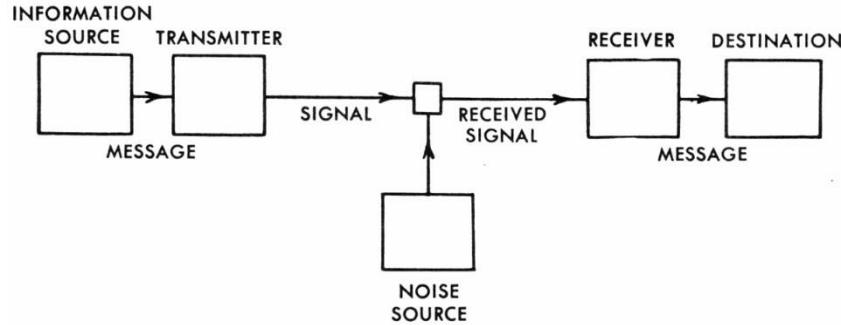
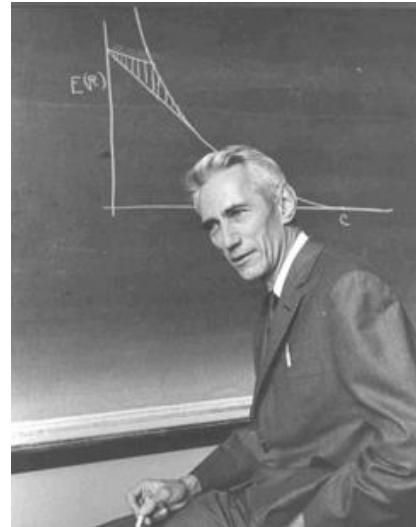


Fig. 1.— Schematic diagram of a general communication system.



Encoder-Decoder System in Communications

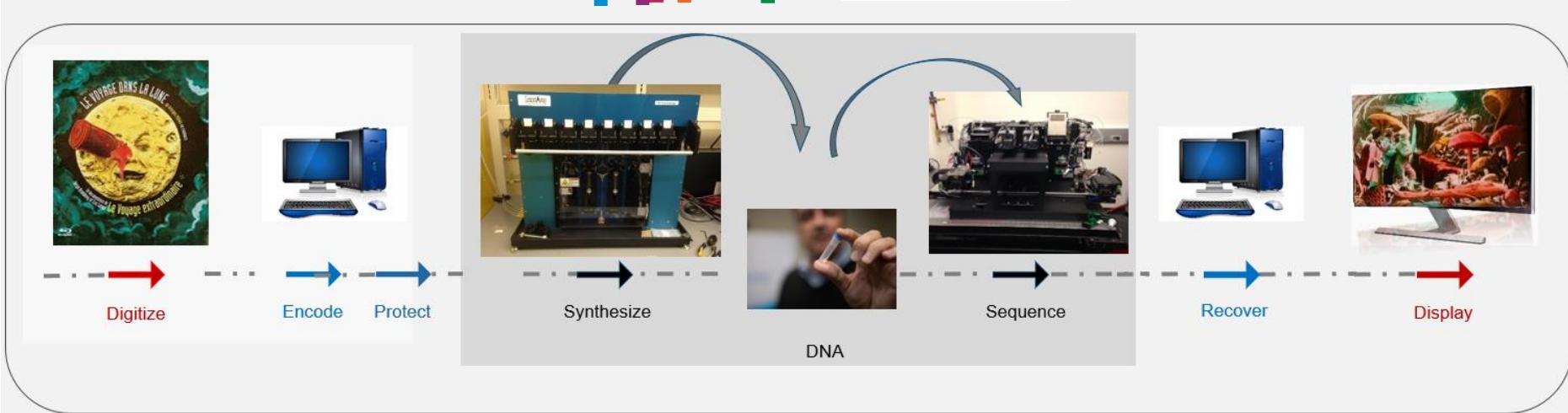
N. Goela, Qualcomm ImpaQt Innovation Award, 2014.

N. Goela, E. Abbe, M. Gastpar, **Polar Codes for Broadcast Channels**, *IEEE Transactions On Information Theory*, vol. 61, no. 2, pp. 758-782, 2015.

N. Goela, M. Raginsky, **Channel Polarization Through The Lens Of Blackwell Measures**, *IEEE Transactions On Information Theory*, vol. 66, no. 10, pp. 6222-6241, 2020.

US-9722651-B2: N. Goela, V. Ekambaram, R. Tandra, J. Soriaga, **An Adaptive Channel Coding System Using Polarization** - Lead Inventor, Qualcomm 5G Technology, 2017.

Encoding and Decoding System in DNA



Digital
Encoder

DNA
Synthesis

DNA
Sequencing

Digital
Decoder

Encoder-Decoder Systems

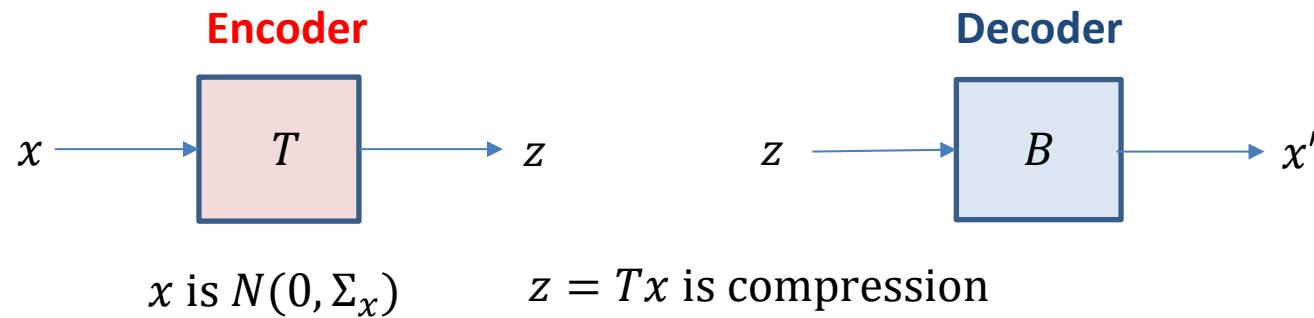
N. Goela, J. Bolot, **Technicolor Visionary Award**, 2016-2017.

H. H. Lee*, R. Kalhor*, N. Goela*, J. Bolot, G. Church, **Terminator-Free Template-Independent Enzymatic DNA Synthesis for Digital Information Storage**, *Nature Communications*, (10), 2383, 2019.

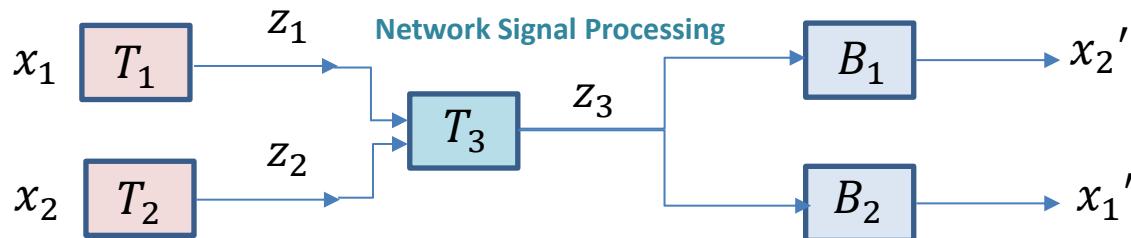
* Equal contribution by authors.

WO-2018-148260-A1: N. Goela, J. Bolot, **Apparatus, Method, and System for Digital Information Storage in DNA - Lead Inventor**, Digital codec technology for DNA molecular storage, 2018.

Encoding and Decoding in Signal Processing Networks



$$B^* = \arg \min_B E[||x - BTx||_2^2] = \Sigma_x T^T (T \Sigma_x T^T)^{-1}$$

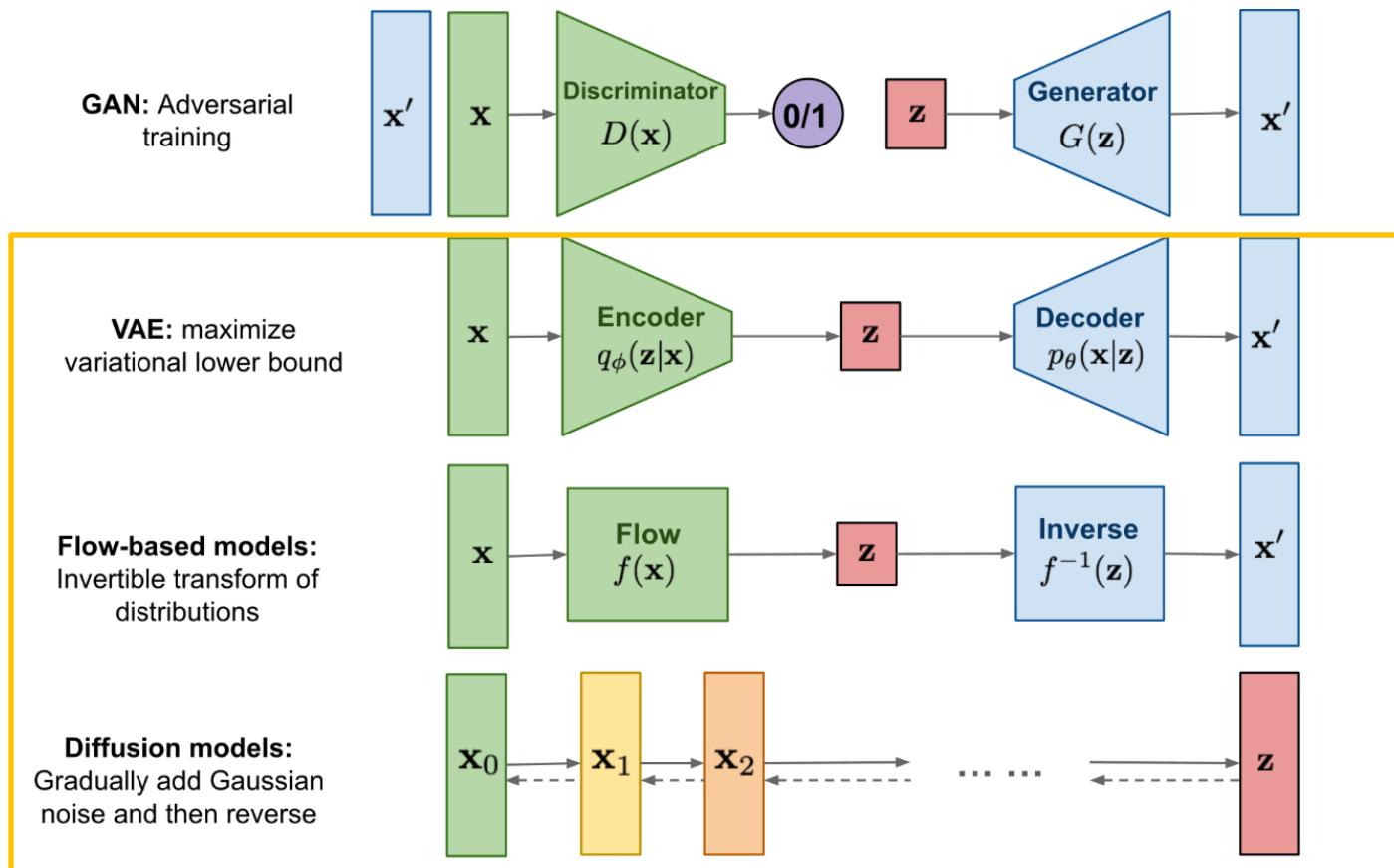


Design T_1, T_2, T_3, B_1, B_2 , compression and estimation matrices

Encoder-Decoder System in Signal Processing

N. Goela, M. Gastpar, **Reduced Dimension Linear Transform Coding of Correlated Signals in Networks**,
IEEE Transactions on Signal Processing, vol. 60, no. 6, pp. 3174-3187, June, 2012.

Encoding and Decoding Systems in Artificial Intelligence, Generative Models



Generative Models

P. Delgosa, N. Goela, **Deep Switch Networks for Generating Discrete Data and Language**, AISTATS-2019, PMLR, vol. 89, no. 12, pp. 3060-3069, Naha, Okinawa, Japan.

Encoding and Decoding in Artificial Intelligence

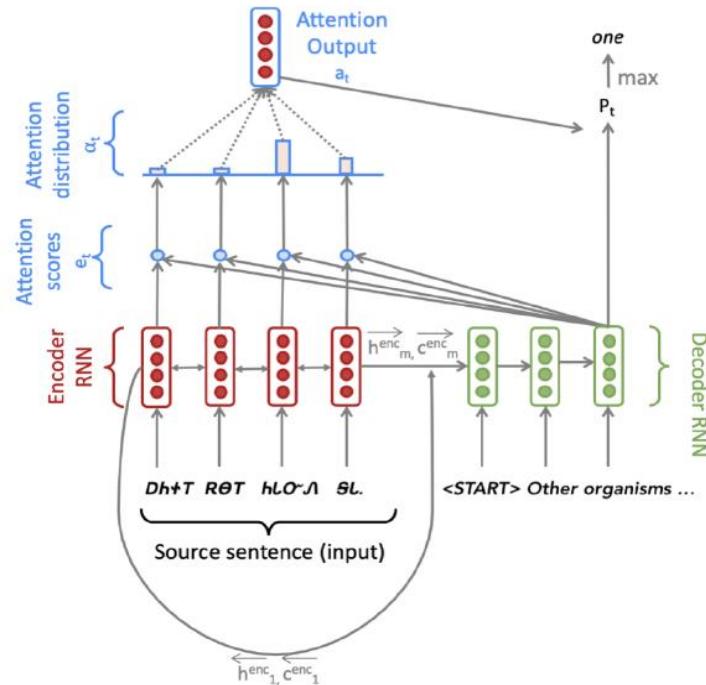


Figure: Neural Machine Translation with Encoder Bidirectional RNN, Decoder RNN, Attention [Manning et al., Notes, 2023]

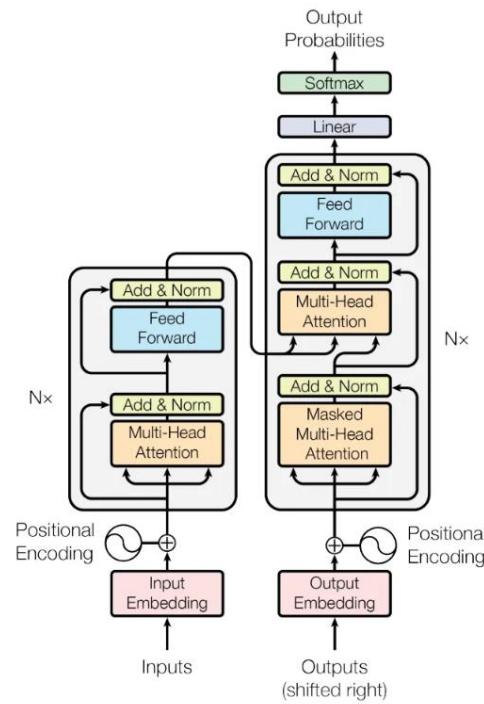


Figure: Transformer BERT Encoder + GPT Decoder, Attention is All You Need [Vaswani et al., 2017]

Encoding and Decoding in VAE, Variational Lower Bounds

**Joint Probability
Distribution**

$$p_{\theta}(x, z) = p_{\theta}(z)p_{\theta}(x|z)$$

Decoder

$$p_{\theta}(x, z) = p_{\theta}(x)p_{\theta}(z|x)$$

**Intractable
Quantities**

$$p_{\theta}(x) = \int p_{\theta}(x, z) dz$$

Encoder

$$p_{\theta}(z|x) = \frac{p_{\theta}(x, z)}{p_{\theta}(x)}$$

$$\begin{aligned} D(q_{\phi}(z|x) || p_{\theta}(z|x)) &= \mathbf{E}_{q_{\phi}(Z|x)} \left[\log \frac{q_{\phi}(Z|x)}{p_{\theta}(Z|x)} \right] = \mathbf{E}_{q_{\phi}(Z|x)} \left[\log \frac{q_{\phi}(z|x)p_{\theta}(x)}{p_{\theta}(x, z)} \right] \\ &= \mathbf{E}_{q_{\phi}(Z|x)} \left[\log \frac{q_{\phi}(Z|x)}{p_{\theta}(x, z)} \right] + \mathbf{E}_{q_{\phi}(Z|x)} [\log p_{\theta}(x)] \end{aligned}$$

$$\log p_{\theta}(x) = \mathbf{E}_{q_{\phi}(Z|x)} \left[\log \frac{p_{\theta}(x, z)}{q_{\phi}(z|x)} \right] + D(q_{\phi}(z|x) || p_{\theta}(z|x))$$

$$L_{\theta, \phi}(x) = \mathbf{E}_{q_{\phi}(Z|x)} \left[\log \frac{p_{\theta}(x, z)}{q_{\phi}(z|x)} \right]$$

Variational Lower Bound

Encoding and Decoding in VAE, Variational Lower Bounds

Joint Probability
Distribution

$$p_{\theta}(x, z) = p_{\theta}(z)p_{\theta}(x|z)$$

Decoder

$$p_{\theta}(x, z) = p_{\theta}(x)p_{\theta}(z|x)$$

Intractable
Quantities

$$p_{\theta}(x) = \int p_{\theta}(x, z) dz$$

Encoder

$$p_{\theta}(z|x) = \frac{p_{\theta}(x, z)}{p_{\theta}(x)}$$

$$\log p_{\theta}(x) \geq L_{\theta, \phi}(x)$$

$$L_{\theta, \phi}(x) = \mathbf{E}_{q_{\phi}(z|x)} \left[\log \frac{p_{\theta}(x, z)}{q_{\phi}(z|x)} \right]$$

$$L_{\theta, \phi}(x) = \mathbf{E}_{q_{\phi}(z|x)} [p_{\theta}(x|z)] - \mathbf{E}_{q_{\phi}(z|x)} \left[\log \frac{q_{\phi}(z|x)}{p_{\theta}(z)} \right]$$

maximize: $L_{\theta, \phi}(x)$

Variational Bounds for Optimizing the Information Bottleneck

Estimating Mutual Information (MINE)

Information Bottleneck Optimization

Inputs: $p(x, y), \beta$

Outputs: $p(z|x), p(z), p(y|z)$ (Self Consistent)

$$\begin{array}{ll} \min_{\substack{p(z|x) \\ Y-X-Z}} & I(X; Z) - \beta I(Z; Y) \\ p(x,y,z)=p(x,y)p(z|x) \end{array}$$

Discrete Case	Gaussian Vector Case	Deep Neural Networks
Discrete Random Variable Z , Z is like “clustering” of X .	Multivariate Gaussian Random Vectors X, Y . Then Z is linear function of X .	Arbitrary Distributions, Trained By Optimizing Variational Lower Bounds.
$p(z x) = \frac{p(z)}{Z(x, \beta)} e^{-\beta D_{KL}[p(y x) p(y z)]}, \quad \forall x, \forall z$	Linear Projection, via Eigenvectors of $\Sigma_{X Y} \Sigma_X^{-1}$	DNN Encoder $p(z x)$, Variational Approximations to: $p(z), p(y z)$
N. Tishby, F. C. Pereira, W. Bialek, The Information Bottleneck Method , Allerton, 1999.	G. Chechik, A. Globerson, N. Tishby, Y. Weiss, Information Bottleneck for Gaussian Variables , Journal of Machine Learning Research, 2005.	A. Alemi, I. Fischer, J. Dillon, K. Murphy, Deep Variational Information Bottleneck , ICLR, 2017.

A Few Information Decompositions

$$X \longrightarrow Y = f(X)$$

$$\begin{aligned} & \max_{\phi, \alpha, \beta} I(\phi(\alpha(X), \beta(X)); f(X)) \\ \text{subject to} \quad & I(\alpha(X); \beta(X)) \leq \Gamma \\ & I(X; \alpha(X), \beta(X)) \leq \Lambda \end{aligned}$$

$$X \longrightarrow Z_1 = \alpha(X)$$

$$X \longrightarrow Z_2 = \beta(X)$$

$$\phi(\alpha(X), \beta(X))$$

Decompose $f(X)$

$$X \longrightarrow \alpha_1(X)$$

$$\phi_1(\alpha_1(X), \beta_1(X))$$

$$X \longrightarrow \beta_1(X)$$

$$\phi(\phi_1(\alpha_1(X), \beta_1(X)), \phi_2(\alpha_2(X), \beta_2(X)))$$

$$X \longrightarrow \alpha_2(X)$$

$$\phi_2(\alpha_2(X), \beta_2(X))$$

$$X \longrightarrow \beta_2(X)$$

Decompose $f(X)$ iteratively

MINE Estimator for Mutual Information

$$D_{KL}(P||Q) = \sup_{T:\mathcal{X} \rightarrow \mathbb{R}} E_P[T(X)] - \log E_Q[e^{T(X)}]$$

Proof (i): $D_{KL}(P||Q) = E_P[T^*(X)] - \log E_Q[e^{T^*(X)}]$

$$T^*(x) = \log \frac{P(x)}{Q(x)} \quad \text{Existence of supremum}$$

Proof (ii): $D_{KL}(P||Q) \geq E_P[T(X)] - \log E_Q[e^{T(X)}]$ for all $T(x)$

Let $G(x) = \frac{Q(x)e^{T(x)}}{E_Q[e^{T(X)}]}$ which is a probability distribution.

$$\text{Then } D_{KL}(P||Q) = E_P[T(X)] + \log E_Q[e^{T(X)}] = D(P||G) \geq 0$$

Appendix: Mathematical Derivations

Continuous Normalized Flows

Applying Jacobi's Formula

$$\frac{d}{dx} \log |\det A(x)| = \text{tr} \left(A(x)^{-1} \frac{dA(x)}{dx} \right)$$

Proof:

$$\begin{aligned}\frac{d}{dx} \log |\det A(x)| &= \frac{1}{\det A(x)} \left[\frac{d}{dx} \det A(x) \right] \\ &= \frac{1}{\det A(x)} \left[\det A(x) \text{tr} \left(A(x)^{-1} \frac{dA(x)}{dx} \right) \right] \\ &= \text{tr} \left(A(x)^{-1} \frac{dA(x)}{dx} \right)\end{aligned}$$

**Jacobi's
Formula**

Continuous Normalized Flows

Jacobi's Formula

$$\frac{d}{dx} \det A(x) = \det A(x) \operatorname{tr} \left(A(x)^{-1} \frac{dA(x)}{dx} \right)$$

$$A(x) = \begin{bmatrix} a_{11}(x) & \cdots & a_{1n}(x) \\ \vdots & \ddots & \vdots \\ a_{n1}(x) & \cdots & a_{nn}(x) \end{bmatrix}$$

Continuous Normalized Flows

Jacobi's Formula

$$\frac{d}{dx} \det A(x) = \det A(x) \operatorname{tr} \left(A(x)^{-1} \frac{dA(x)}{dx} \right)$$

$$\det(\exp(M(x))) = \exp [\operatorname{tr}(M(x))]$$

Proof: $\frac{d}{dx} \det A(x) = \frac{d}{dx} \det(\exp(\log A(x)))$

$$= \frac{d}{dx} \exp [\operatorname{tr}(\log A(x))]$$
$$= \exp [\operatorname{tr}(\log A(x))] \frac{d}{dx} [\operatorname{tr}(\log A(x))]$$
$$= \exp [\operatorname{tr}(\log A(x))] \operatorname{tr} \left[\frac{d}{dx} (\log A(x)) \right]$$
$$= \det A(x) \operatorname{tr} \left(A(x)^{-1} \frac{dA(x)}{dx} \right)$$


Finite and Continuous Normalizing Flows

Neural ODEs

Consider:

$$y = f(x), \quad Y = f(X)$$

Theorem:

$$\log p_Y(f(x)) = \log p_X(x) - \log \left| \det \left[\frac{\partial f}{\partial x} \right] \right|$$

Finite Normalizing Flows

Danilo Rezende, Shakir Mohamed, **Variational Inference with Normalizing Flows**, *Proceedings of the 32nd International Conference on Machine Learning*, PMLR 37:1530-1538, 2015.

Consider:

$$\frac{dz(t)}{dt} = f(z(t), t)$$

Theorem:

$$\frac{d \log p(z(t))}{dt} = -\text{tr} \left[\frac{\partial f}{\partial z} \right]$$

Continuous Normalizing Flows: Neural ODEs

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, David K. Duvenaud, **Neural Ordinary Differential Equations**, *Advances in Neural Information Processing Systems*, vol. 31, 2018.

Proof of Theorem: Neural ODEs

Consider:

$$\frac{dz(t)}{dt} = f(z(t), t)$$

Theorem:

$$\frac{d \log p(z(t))}{dt} = -\text{tr} \left[\frac{\partial f}{\partial z} \right]$$

$$z(t) = \begin{pmatrix} z_1(t) \\ \vdots \\ z_n(t) \end{pmatrix}$$

$$f(z(t), t) = \begin{pmatrix} f_1(z(t), t) \\ \vdots \\ f_n(z(t), t) \end{pmatrix}$$

$$\frac{\partial f}{\partial z} = \begin{bmatrix} \frac{\partial f_1}{\partial z_1} & \dots & \frac{\partial f_1}{\partial z_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial z_1} & \dots & \frac{\partial f_n}{\partial z_n} \end{bmatrix}$$

$$\begin{aligned} z(t + \epsilon) &= \int_t^{t+\epsilon} f(z(\tau), \tau) d\tau \\ &= T_\epsilon(z(t)) \\ &= z(t) + \epsilon f(z(t), t) + o(\epsilon) \end{aligned}$$

$$\left[\frac{\partial T_\epsilon(z(t))}{\partial z} \right] = I_n + \epsilon \left[\frac{\partial f}{\partial z} \right] + o(\epsilon)$$

$$\lim_{\epsilon \rightarrow 0} \left[\frac{\partial T_\epsilon(z(t))}{\partial z} \right]^{-1} = I_n$$

$$\lim_{\epsilon \rightarrow 0} \frac{\partial}{\partial \epsilon} \left[\frac{\partial T_\epsilon(z(t))}{\partial z} \right] = \left[\frac{\partial f}{\partial z} \right]$$

Proof of Theorem: Neural ODEs

Simplified Derivation using Jacobi's Formula

$$\begin{aligned}\frac{d \log p(z(t))}{dt} &= \lim_{\epsilon \rightarrow 0} \left(\frac{\log p(z(t+\epsilon)) - \log p(z(t))}{\epsilon} \right) \\&= -\lim_{\epsilon \rightarrow 0} \left(\frac{\log \left| \det \left[\frac{\partial T_\epsilon(z(t))}{\partial z} \right] \right|}{\epsilon} \right) \\&= -\lim_{\epsilon \rightarrow 0} \left(\frac{\partial}{\partial \epsilon} \log \left| \det \left[\frac{\partial T_\epsilon(z(t))}{\partial z} \right] \right| \right) \\&= -\lim_{\epsilon \rightarrow 0} \left(\text{tr} \left[\left[\frac{\partial T_\epsilon(z(t))}{\partial z} \right]^{-1} \frac{\partial}{\partial \epsilon} \left[\frac{\partial T_\epsilon(z(t))}{\partial z} \right] \right] \right) \\&= -\text{tr} \left[\frac{\partial f}{\partial z} \right]\end{aligned}$$



Definition

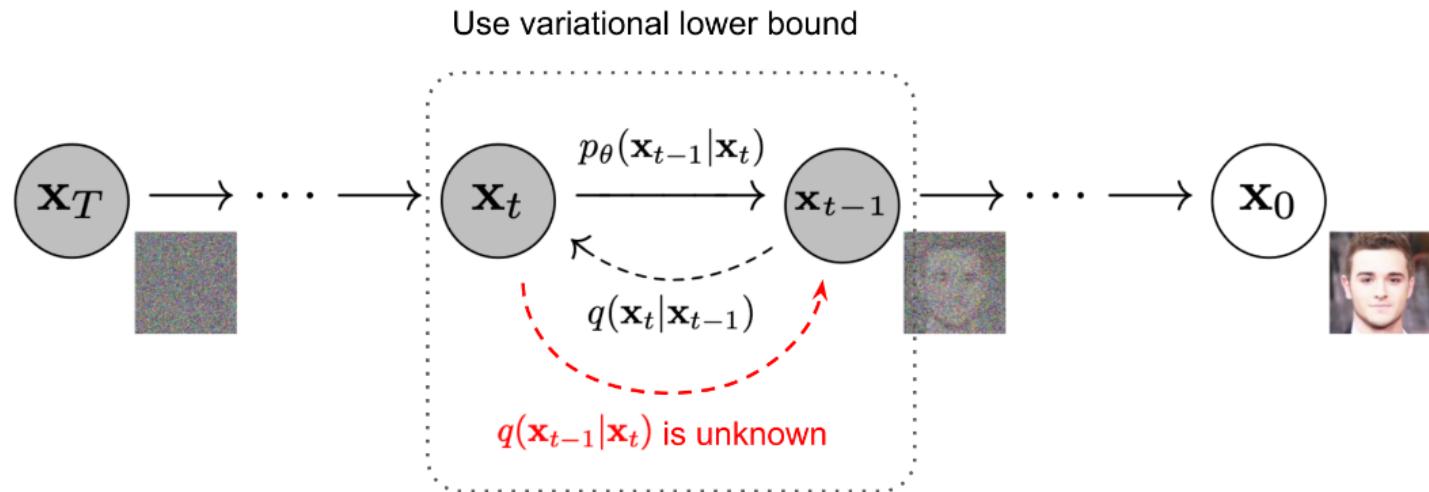
Change of
Variables Rule

L'Hôpital's Rule

Jacobi's
Formula

Taking Limits

Variational Bounds for Diffusion Models



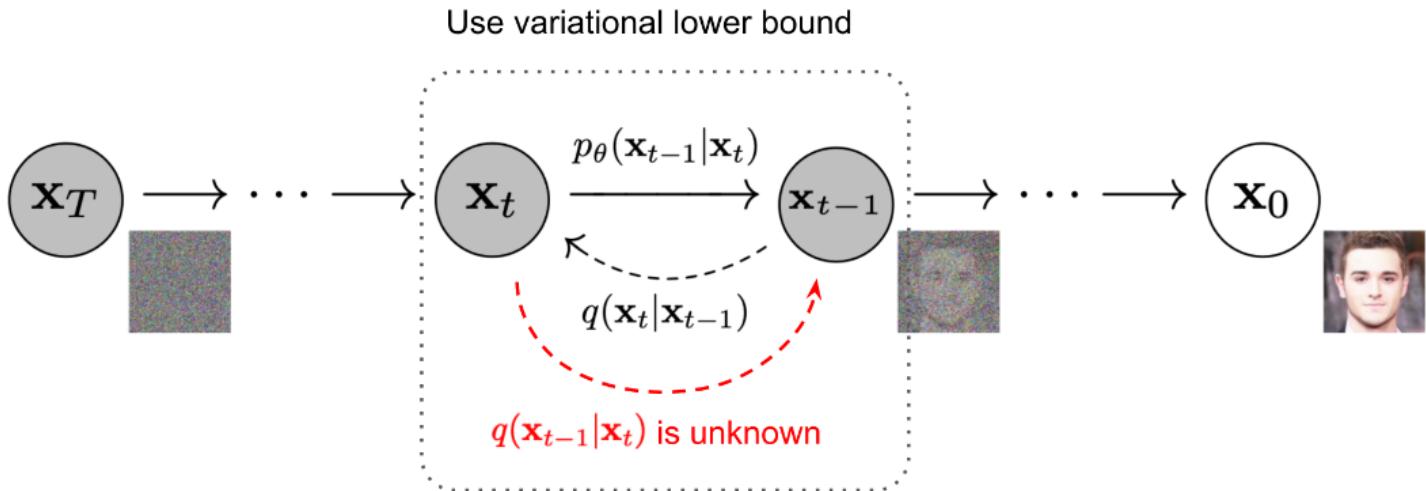
ODEs to Stochastic Differential Equations

D. Duvenaud et al., **When should we make our models continuous in time?**, Stochastic Differential Equations, UC Berkeley Colloquium, 2022.

Denoising Diffusion Probabilistic Models

J. Ho, A. Jain, P. Abbeel, **Denoising Diffusion Probabilistic Models**, *Advances in Neural Information Processing Systems*, 2020.

Variational Bounds for Diffusion Models



Forward Diffusion

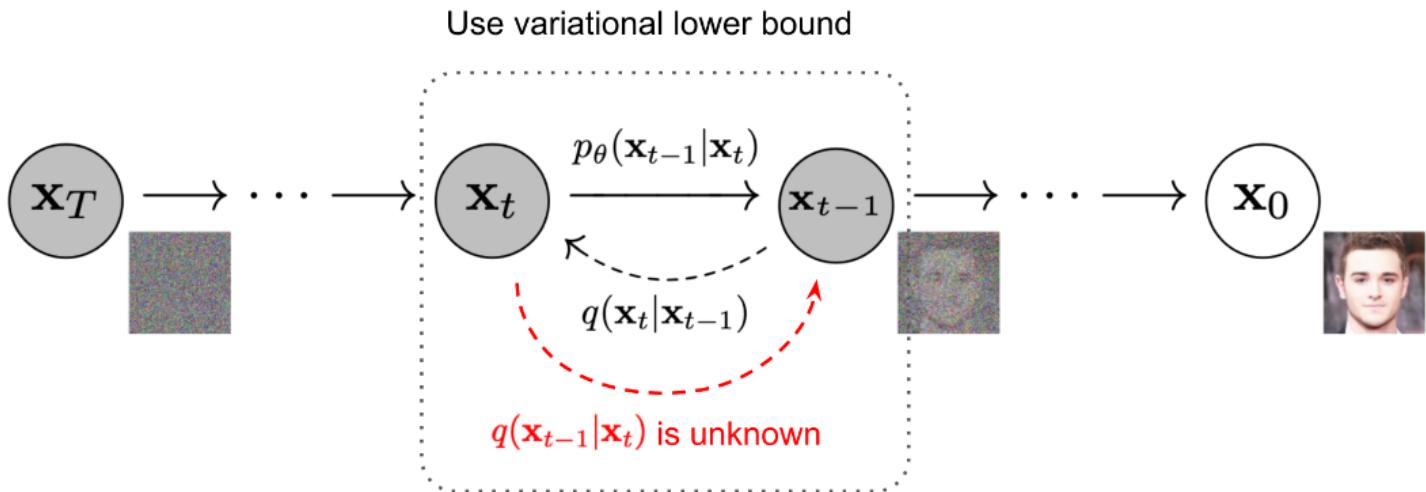
$$q\left(x_t \mid x_{t-1}\right)=N\left(x_t ; x_{t-1} \sqrt{1-\beta_t}, \beta_t I\right)$$

$$x_t = x_{t-1} \sqrt{1 - \beta_t} + \sqrt{\beta_t} \epsilon_{t-1} \text{ where } \epsilon_{t-1} \sim N(0, I)$$

Reverse Diffusion

$$p_\theta(x_{t-1} | x_t) = N(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$$

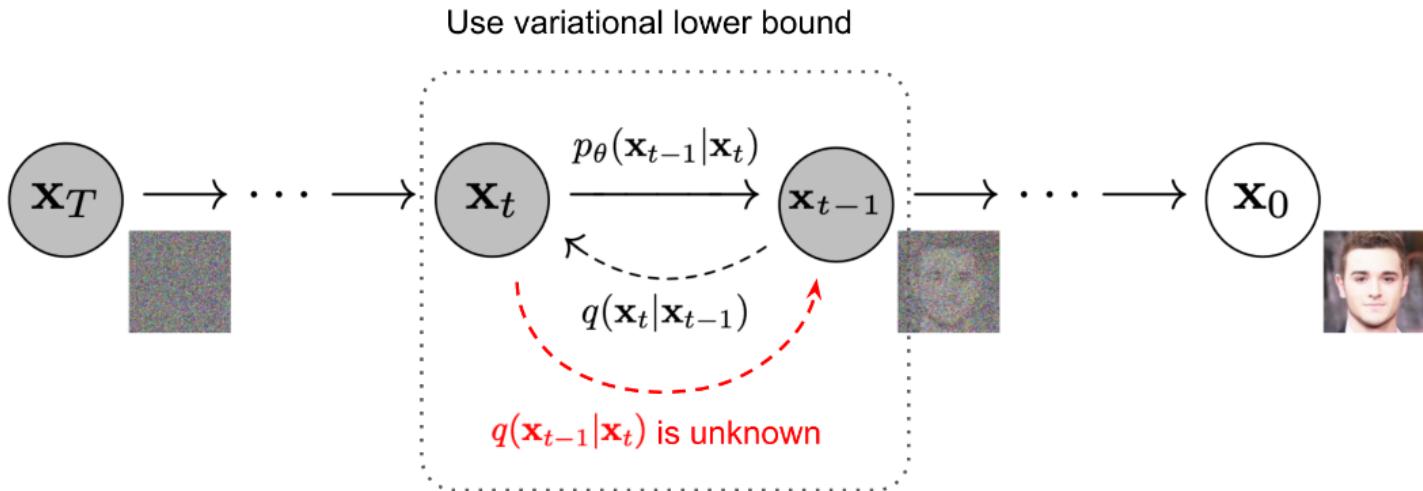
Variational Bounds for Diffusion Models



Conditional KL-Divergence

$$\begin{aligned} KL(q(x_{1:T}|x_0) || p_\theta(x_{1:T}|x_0) | x_0) &= \mathbf{E}_{q(x_0)} \mathbf{E}_{q(x_{1:T}|x_0)} \left[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{1:T}|x_0)} \right] \\ &= \mathbf{E}_{q(x_{0:T})} \left[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})} \right] + \mathbf{E}_{q(x_{0:T})} [\log p_\theta(x_0)] \end{aligned}$$

Variational Bounds for Diffusion Models



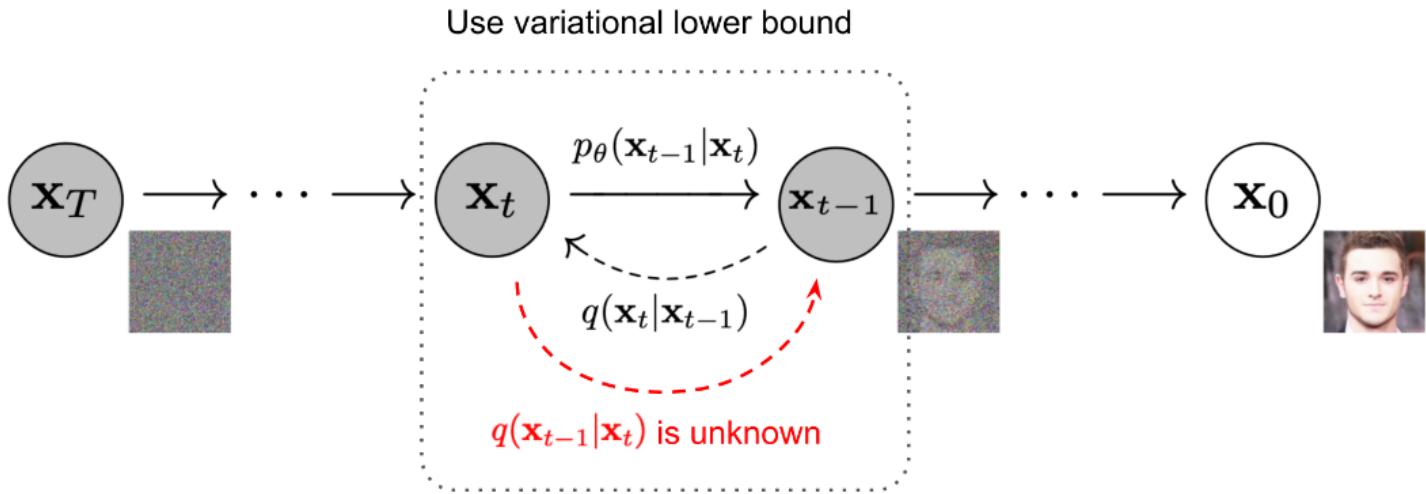
Conditional KL-Divergence

$$\begin{aligned} KL(q(x_{1:T}|x_0) || p_\theta(x_{1:T}|x_0) | x_0) &= \mathbf{E}_{q(x_0)} \mathbf{E}_{q(x_{1:T}|x_0)} \left[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{1:T}|x_0)} \right] \\ &= \mathbf{E}_{q(x_{0:T})} \left[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})} \right] + \mathbf{E}_{q(x_{0:T})} [\log p_\theta(x_0)] \end{aligned}$$

$$-\mathbf{E}_{q(x_0)} \log p_\theta(x_0) = \mathbf{E}_{q(x_{0:T})} \left[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})} \right] - KL(q(x_{1:T}|x_0) || p_\theta(x_{1:T}|x_0) | x_0)$$

$$-\mathbf{E}_{q(x_0)} \log p_\theta(x_0) \leq \mathbf{E}_{q(x_{0:T})} \left[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})} \right]$$

Variational Bounds for Diffusion Models



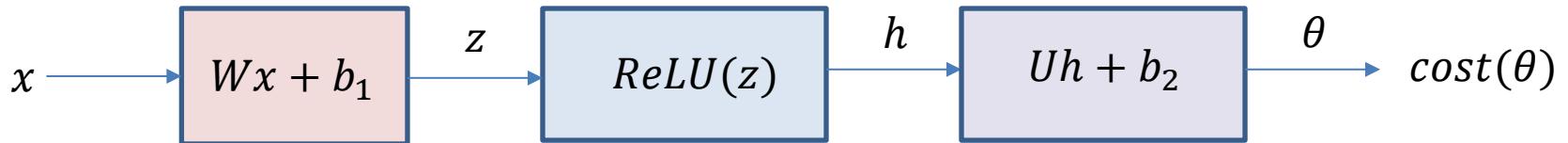
$$-\mathbf{E}_{q(x_0)} \log p_\theta(x_0) \leq \mathbf{E}_{q(x_{0:T})} \left[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})} \right]$$

Minimize Cross-Entropy

$$\text{minimize: } \mathbf{E}_{q(x_{0:T})} \left[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})} \right]$$

Simple Re-Derivation Similar to VAE Derivation

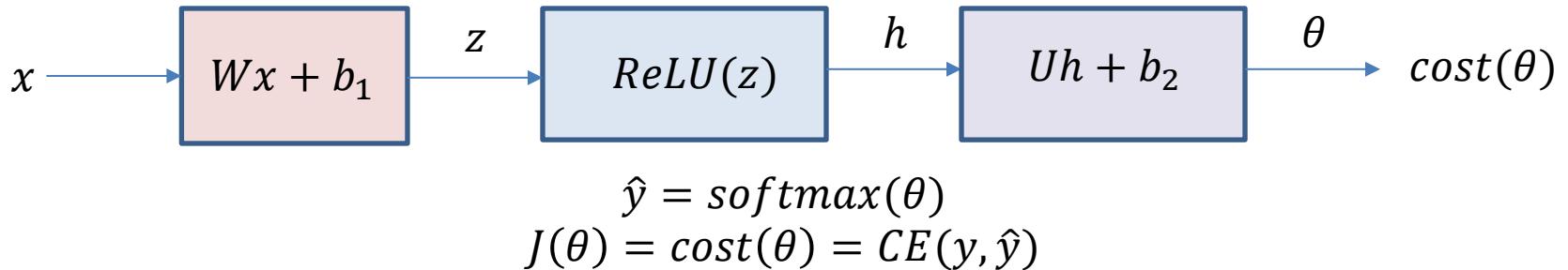
Neural Network with 1 Hidden Layer



$$\begin{aligned}\hat{y} &= \text{softmax}(\theta) \\ J(\theta) &= \text{cost}(\theta) = CE(y, \hat{y})\end{aligned}$$

$$\begin{aligned}\frac{d\text{cost}(\theta)}{d\theta} &= \frac{dJ(\theta)}{d\theta} \\ &= \frac{dJ(\theta)}{d\hat{y}} \frac{d\hat{y}}{d\theta} = \frac{d(CE(y, \hat{y}))}{d\hat{y}} \frac{d(\text{softmax}(\theta))}{d\theta} \\ &= \left[\frac{d(CE(y, \hat{y}))}{d\hat{y}_1} \frac{d(CE(y, \hat{y}))}{d\hat{y}_2} \right] \begin{bmatrix} \frac{d(\text{softmax}(\theta))}{\theta_1} & \frac{d(\text{softmax}(\theta))}{\theta_2} \end{bmatrix} \\ &= \left[\frac{-y_1}{\hat{y}_1} \frac{-y_2}{\hat{y}_2} \right] \begin{bmatrix} \hat{y}_1(1 - \hat{y}_1) & -\hat{y}_1\hat{y}_2 \\ -\hat{y}_1\hat{y}_2 & \hat{y}_2(1 - \hat{y}_2) \end{bmatrix} \\ &= \begin{bmatrix} -y_1(1 - \hat{y}_1) + y_2\hat{y}_1 & y_1\hat{y}_2 - y_2(1 - \hat{y}_2) \end{bmatrix} \\ &= \begin{bmatrix} -y_1 + (1 - y_2)\hat{y}_1 + y_2\hat{y}_1 & -y_2(1 - \hat{y}_2) + y_1\hat{y}_2 \end{bmatrix} \\ &= (\hat{y} - y)^T\end{aligned}$$

Neural Network with 1 Hidden Layer



$$\frac{dJ(\theta)}{dW} = \frac{dJ(\theta)}{dz} \frac{dz}{dW} = \left[\frac{dJ(\theta)}{dz} \right]^T x^T$$

$$\frac{dJ(\theta)}{db_1} = \frac{dJ(\theta)}{dz} \frac{dz}{db_1} = \left[\frac{dJ(\theta)}{dz} \right]^T$$

$$\frac{dJ(\theta)}{dU} = \frac{dJ(\theta)}{d\theta} \frac{d\theta}{dU} = \left[\frac{dJ(\theta)}{d\theta} \right]^T h^T$$

$$\frac{dJ(\theta)}{db_2} = \frac{dJ(\theta)}{d\theta} \frac{d\theta}{db_2} = \left[\frac{dJ(\theta)}{d\theta} \right]^T$$

$$\begin{aligned} \left[\frac{dJ(\theta)}{dz} \right] &= \frac{dJ(\theta)}{d\theta} \frac{d\theta}{dh} \frac{d(h)}{dz} \\ &= (\hat{y} - y)^T U \circ \text{RELU}'(z) \end{aligned}$$

$$\left[\frac{dJ(\theta)}{d\theta} \right] = (\hat{y} - y)^T$$

Part II

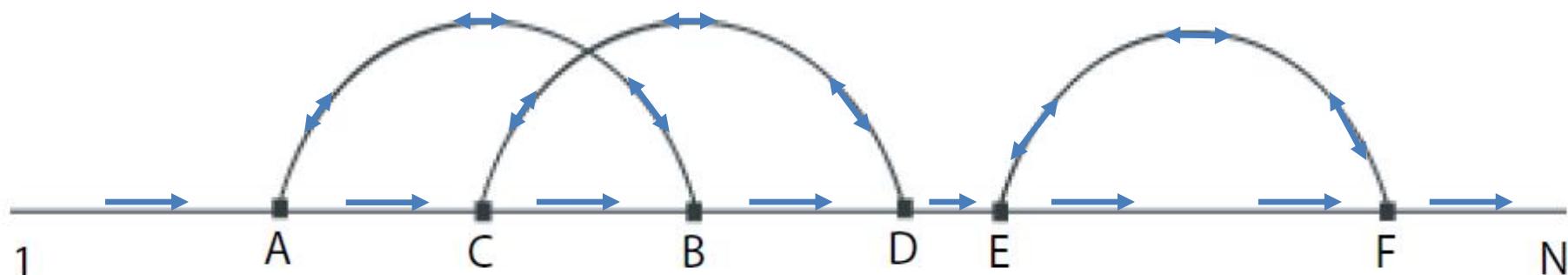
Computer Vision Systems

Visual Hull Motion Capture Multi-Camera System	 The logo for MIT Computer Science and Artificial Intelligence Laboratory (CSAIL) features a stylized orange and white geometric structure resembling a city skyline or abstract buildings, with the text "MIT CSAIL" below it.	Academia
Aerial City-Wide Persistent Surveillance	 The logo for Lincoln Laboratory, Massachusetts Institute of Technology, features a blue square icon with a grid pattern next to the text "LINCOLN LABORATORY" and "MASSACHUSETTS INSTITUTE OF TECHNOLOGY".	Government
Scalable Multi-Projector Displays Using Multi- Camera Feedback	 The logo for Scalable Display Technologies features the word "scalable" in a lowercase sans-serif font, followed by a graphic element consisting of overlapping colored rectangles (red, yellow, green), and the text "DISPLAY TECHNOLOGIES" in a smaller sans-serif font below.	Industry

Visual Hull Motion Graphs

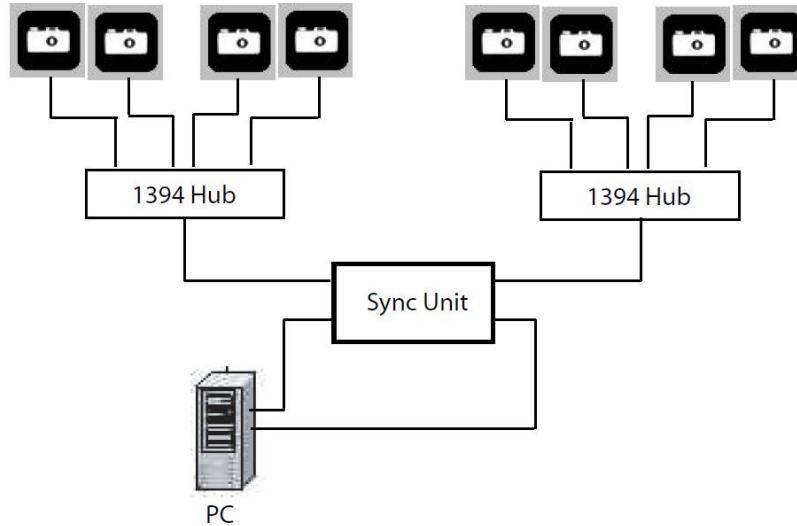


3D Shape, Multiview Motion



Simple and Complex Loops,
Dynamic Programming

Visual Hull Motion Capture System



Synchronized Camera Setup

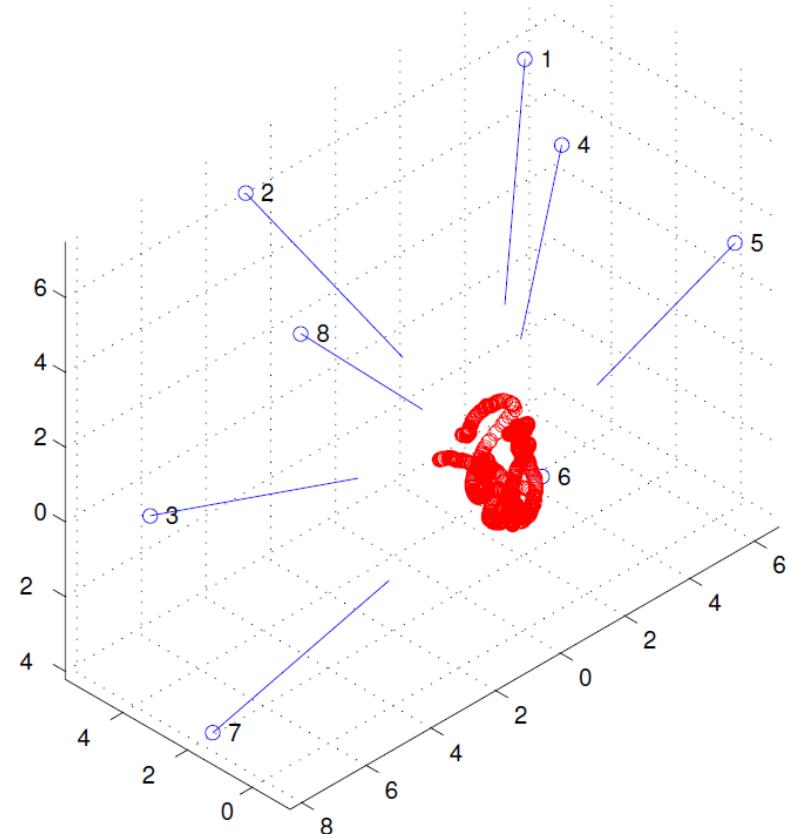
8 DragonFly Cameras, Point Grey Research Inc.

640x480x3 Images, 15 fps.

4-GB RAM on PC.

Store gray scale images, interpolate afterward.

Store Intel JPEG compressed 6:1 images in RAM.



Multiview Camera Calibration

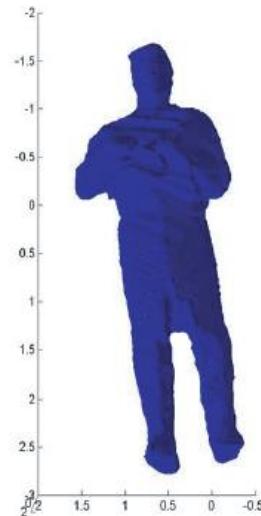
$$\begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix} = \frac{1}{z} K [R \ t] \begin{bmatrix} M \\ 1 \end{bmatrix}$$

Polyhedral Visual Hull Algorithm

Multi-view 2D Inputs, Calibrated Cameras

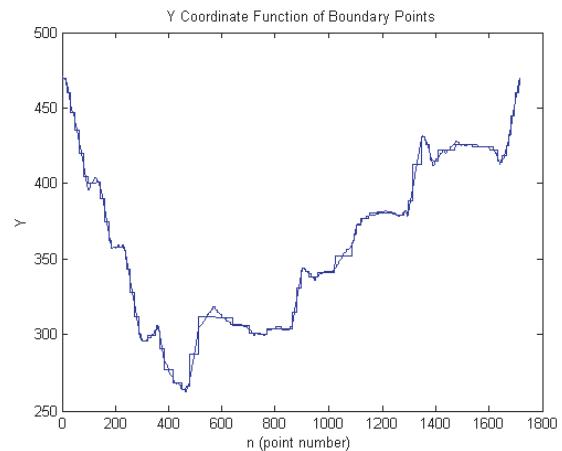
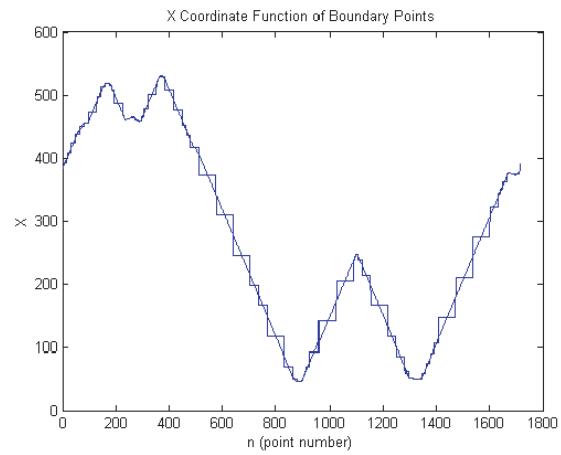
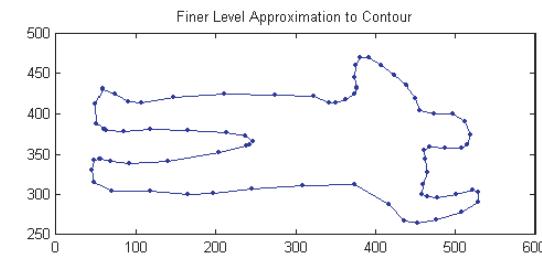
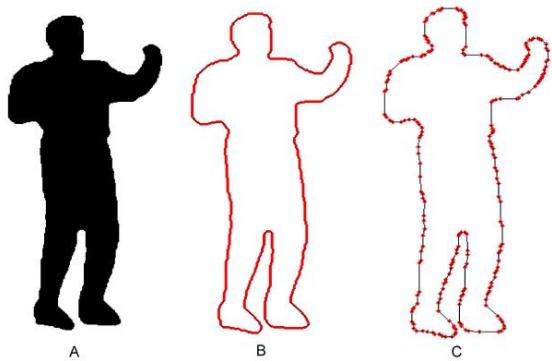


3D Mesh Output



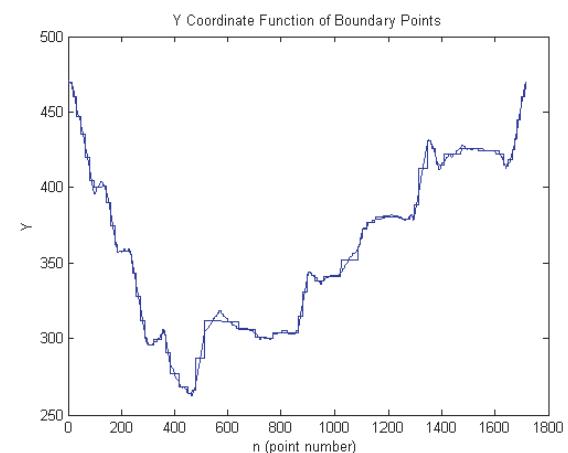
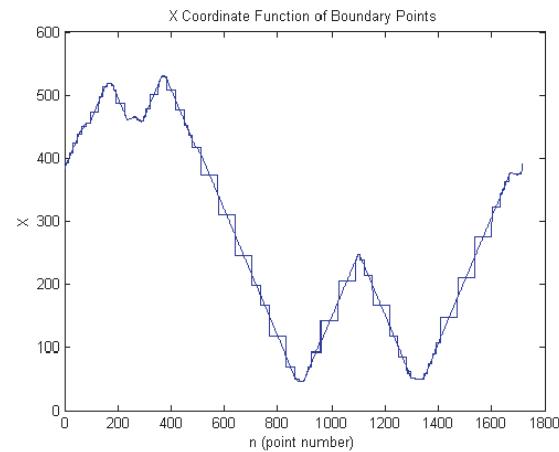
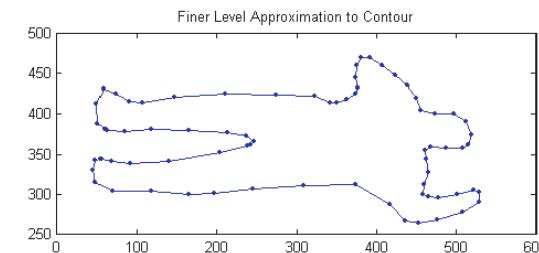
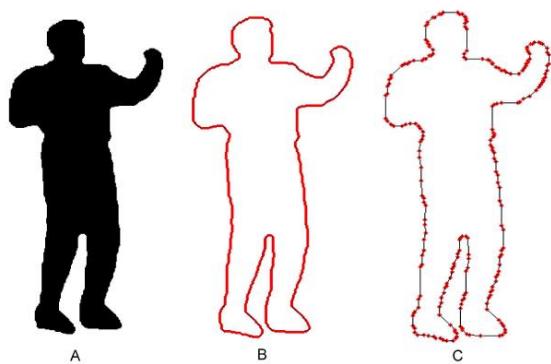
2D Contour Approximation, 2D Shape Wavelets

74-Point Contour Approximation,
From 1716-Point Dense Contour

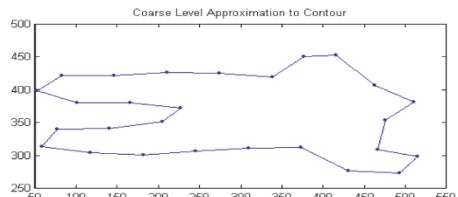


2D Contour Approximation, 2D Shape Wavelets

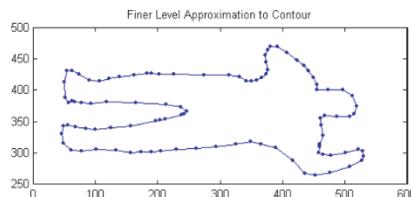
74-Point Contour Approximation,
From 1716-Point Dense Contour



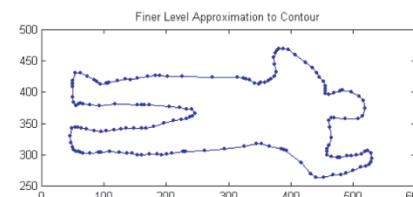
27 Point Contour



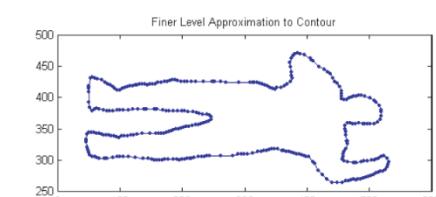
101 Point Contour



171 Point Contour

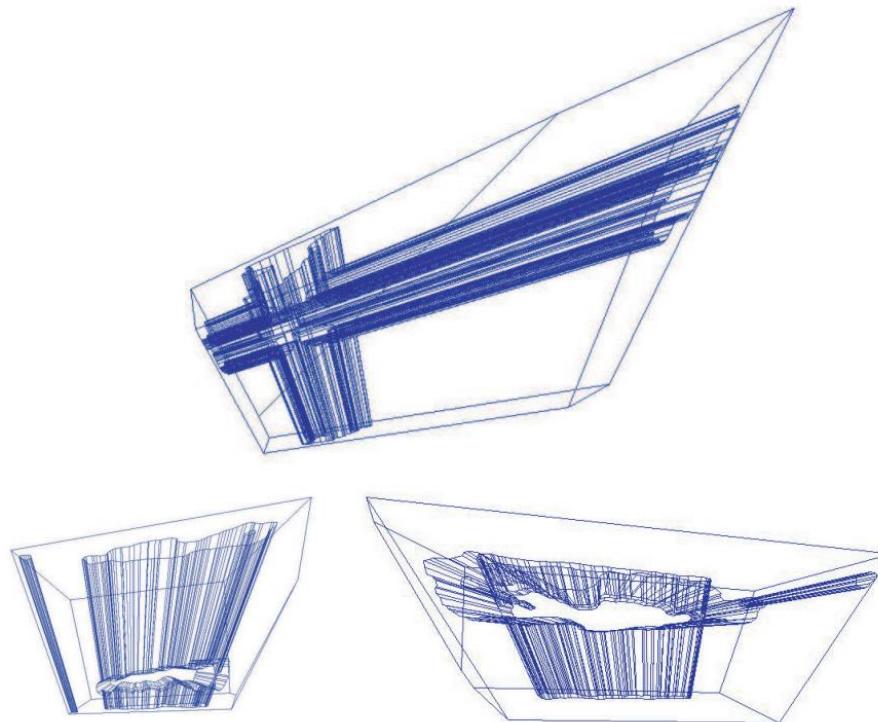


330 Point Contour

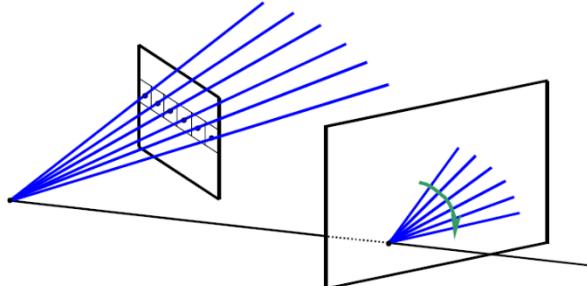


Visual Hull = Intersection Of 3D Silhouette Cones

$$vhull_K = \bigcap_{k \in [K]} C_k$$



Contour Rays Project To Epipolar Rays

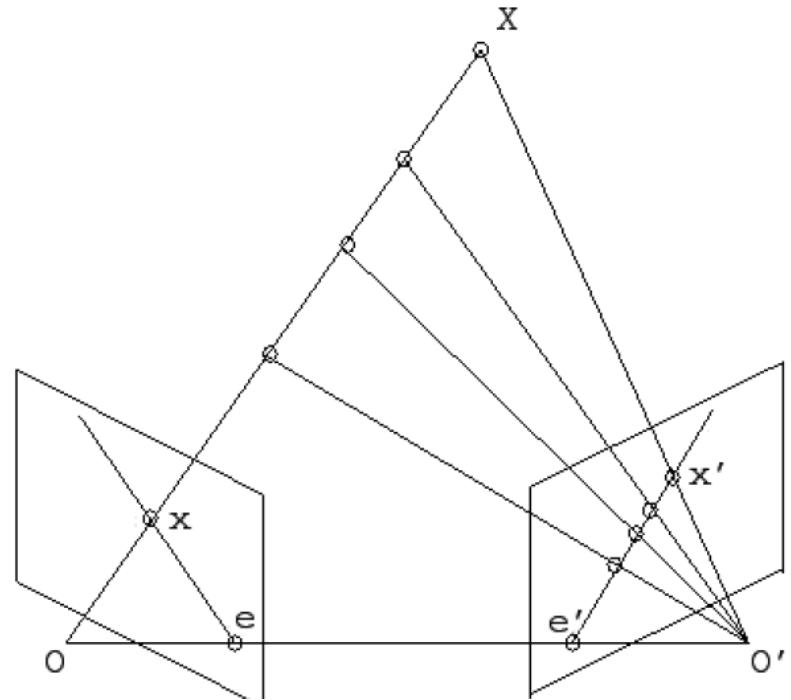


Finding Contour Rays

$$\begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix} = \frac{1}{z} K [R \ t] \begin{bmatrix} M \\ 1 \end{bmatrix}$$

$$M = R^T K^{-1} z \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix} - R^T t$$

$$camera_{eye} = -R^T t$$

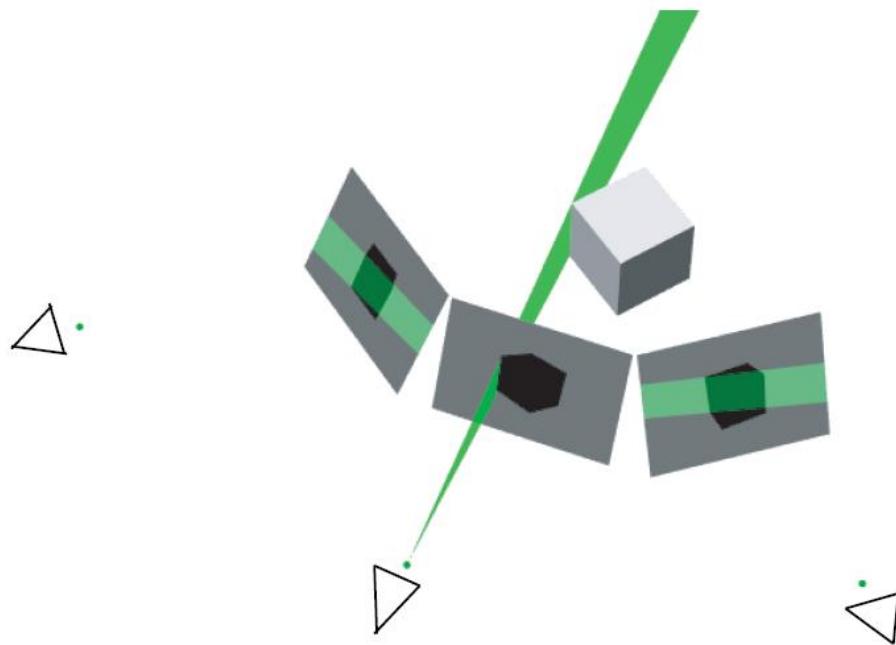


Epipolar Constraint

Points on \overrightarrow{OX} project to points on $\overrightarrow{e'x'}$

Silhouette Face Projects to Epipolar Face

$$vhull_K = \bigcap_{k \in [K]} C_k$$



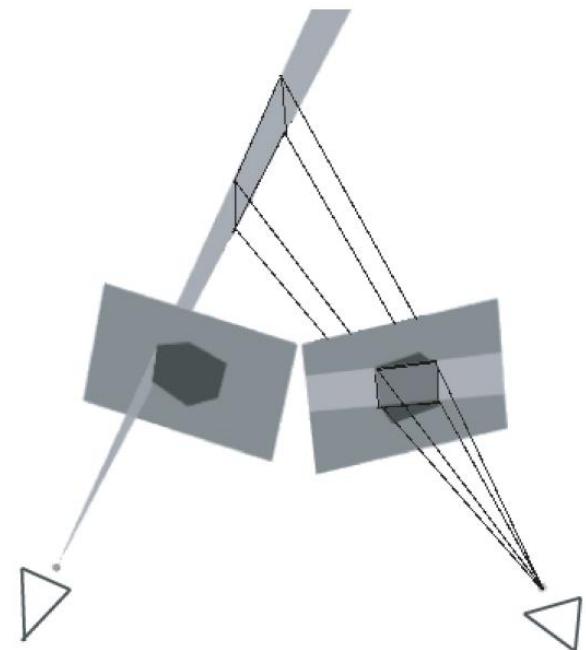
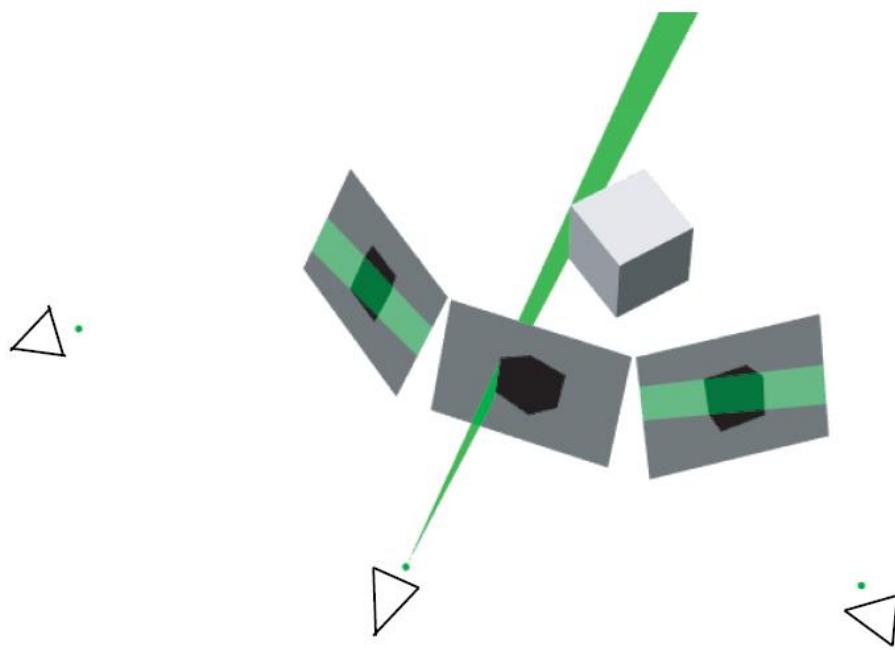
Matusik et al. Polyhedral Visual Hulls For Real-Time Rendering,
Eurographics Workshop on Rendering Techniques, 2001

Matusik et al. Image-Based Visual Hulls, *Siggraph*, 2000

Silhouette Face Projects to Epipolar Face

$$vhull_K = \bigcap_{k \in [K]} C_k$$

Reduce 3D Intersection of
Silhouette Cones to 2D
Polygon Intersections

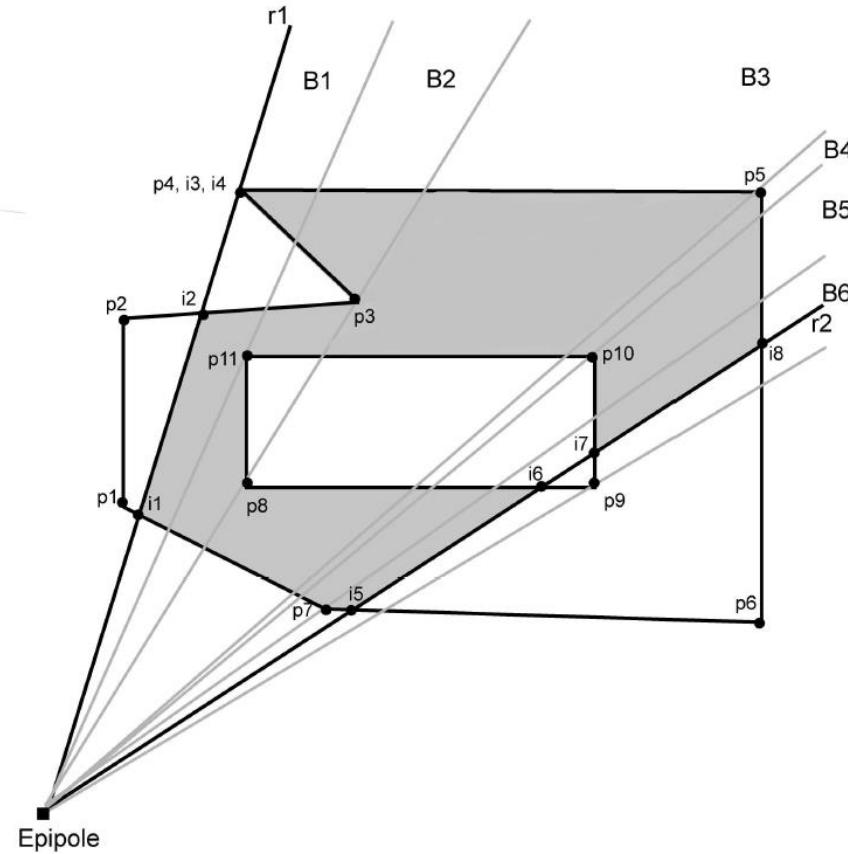
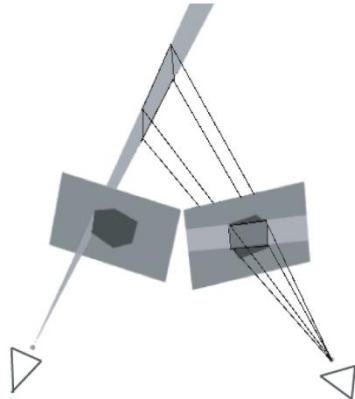


Matusik et al. Polyhedral Visual Hulls For Real-Time Rendering,
Eurographics Workshop on Rendering Techniques, 2001

Matusik et al. Image-Based Visual Hulls, Siggraph, 2000

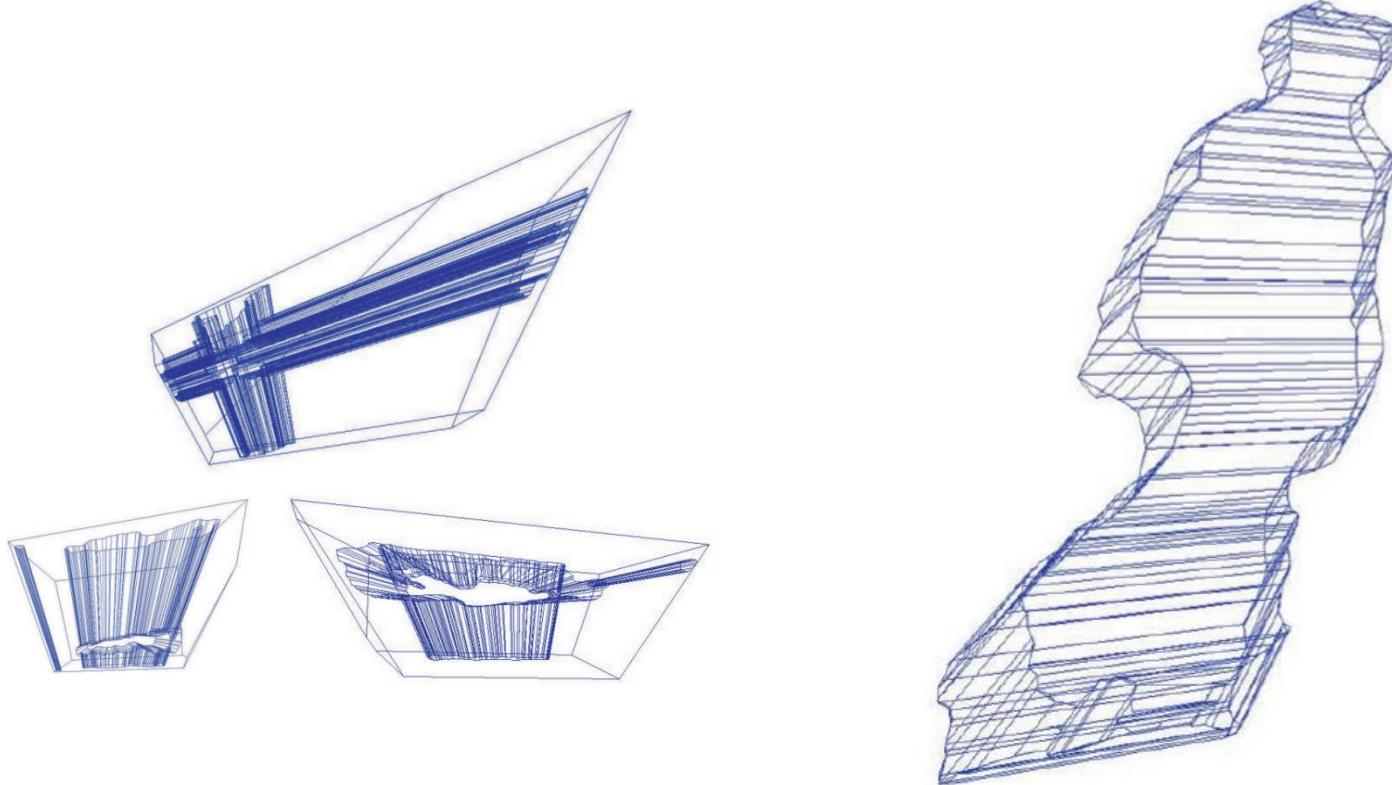
Epipolar Face and Silhouette 2D-Intersection

Reduce 3D Intersection of
Silhouette Cones to 2D
Polygon Intersections



Visual Hull With Two Camera Views

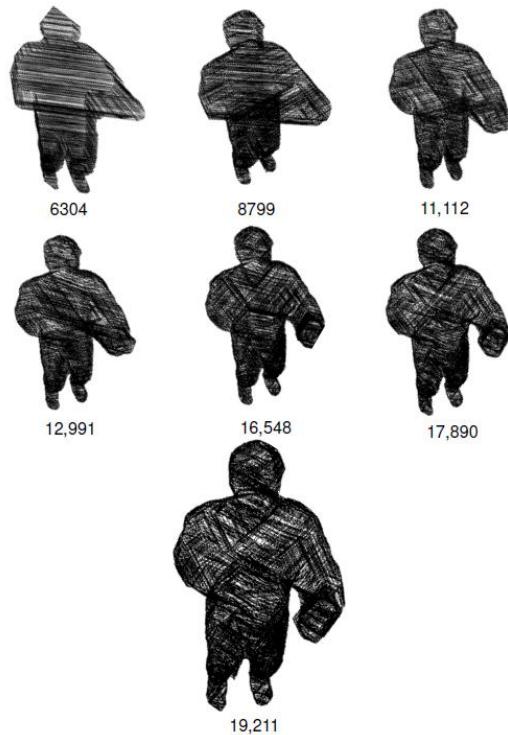
$$vhull_{K=2} = C_1 \cap C_2$$



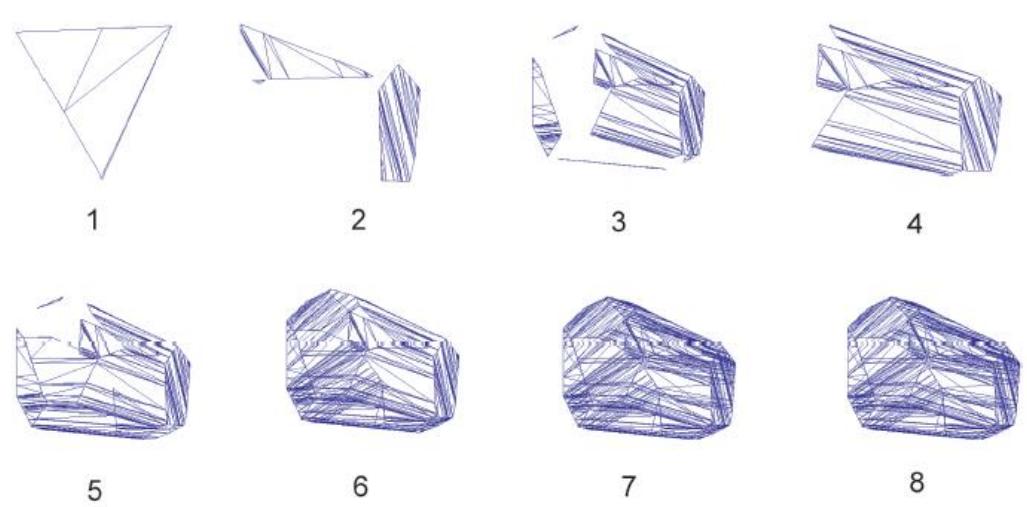
Visual Hull With Eight Camera Views

$$vhull_{K=8} = \bigcap_{k \in [8]} C_k$$

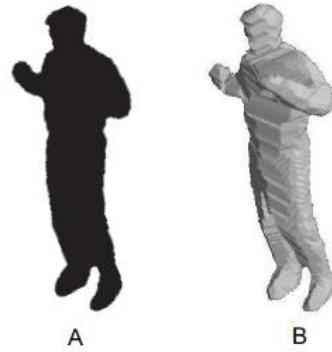
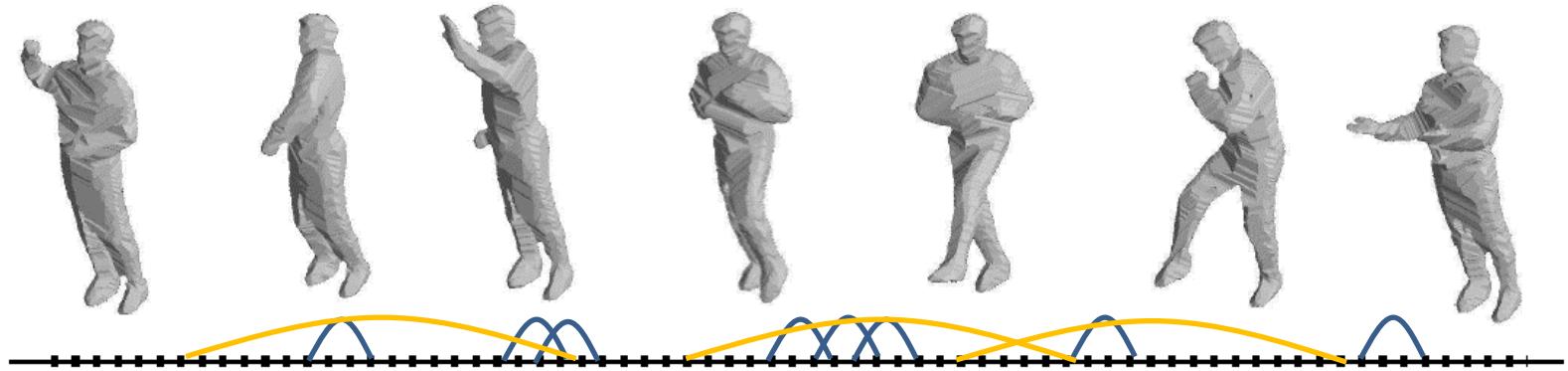
Triangle Count in 3D Output Mesh



Each 3D Silhouette Cone Contributors 3D Surfaces



Visual Hull Motion Graphs



Inferring 3D Pose From 2D Silhouettes,
Chamfer Shape Distance

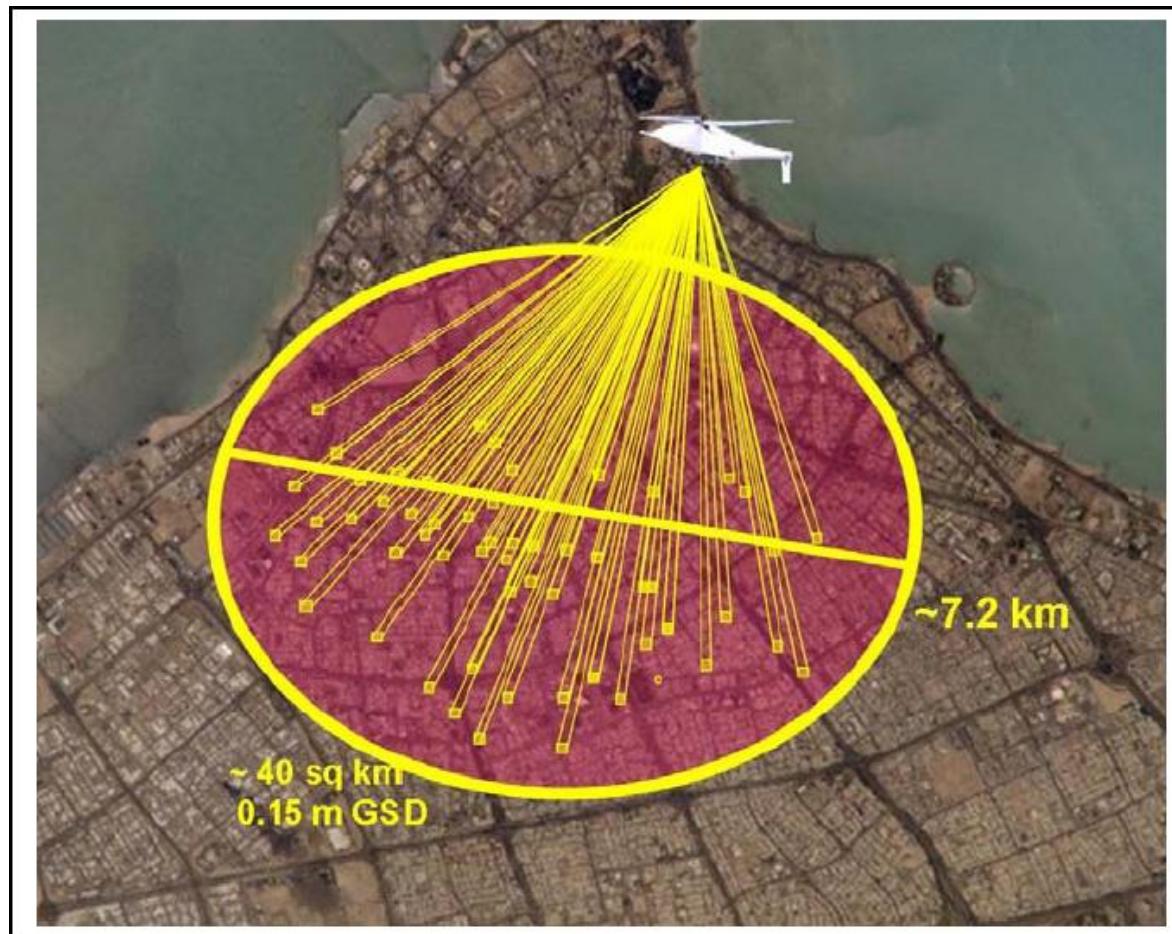
$$D_{chamfer}(U, V) = \frac{1}{N} \sum_{u_i \in U} \min_{v_j \in V} \|u_i - v_j\|$$

$$U = \{\mu_i\}_{i \in [N]} \quad V = \{v_j\}_{j \in [M]}$$

Computer Vision for Aerial City-Wide Persistent Surveillance



Computer Vision for Aerial City-Wide Persistent Surveillance



2D Image Registration

MathWorks Image Registration

<https://www.mathworks.com/discovery/image-registration.html>



Registering aerial photos using point mapping.

2D Image Registration

MathWorks Image Registration

<https://www.mathworks.com/discovery/image-registration.html>

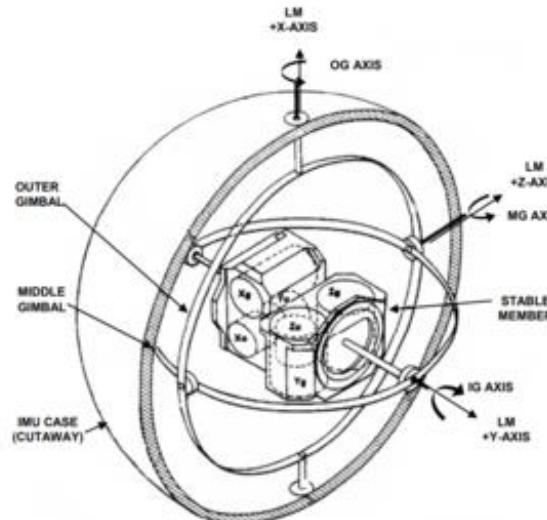


Registering aerial photos using point mapping.

3D Camera Parameters

Inertial Measurement Unit (IMU)

On Board Airplane

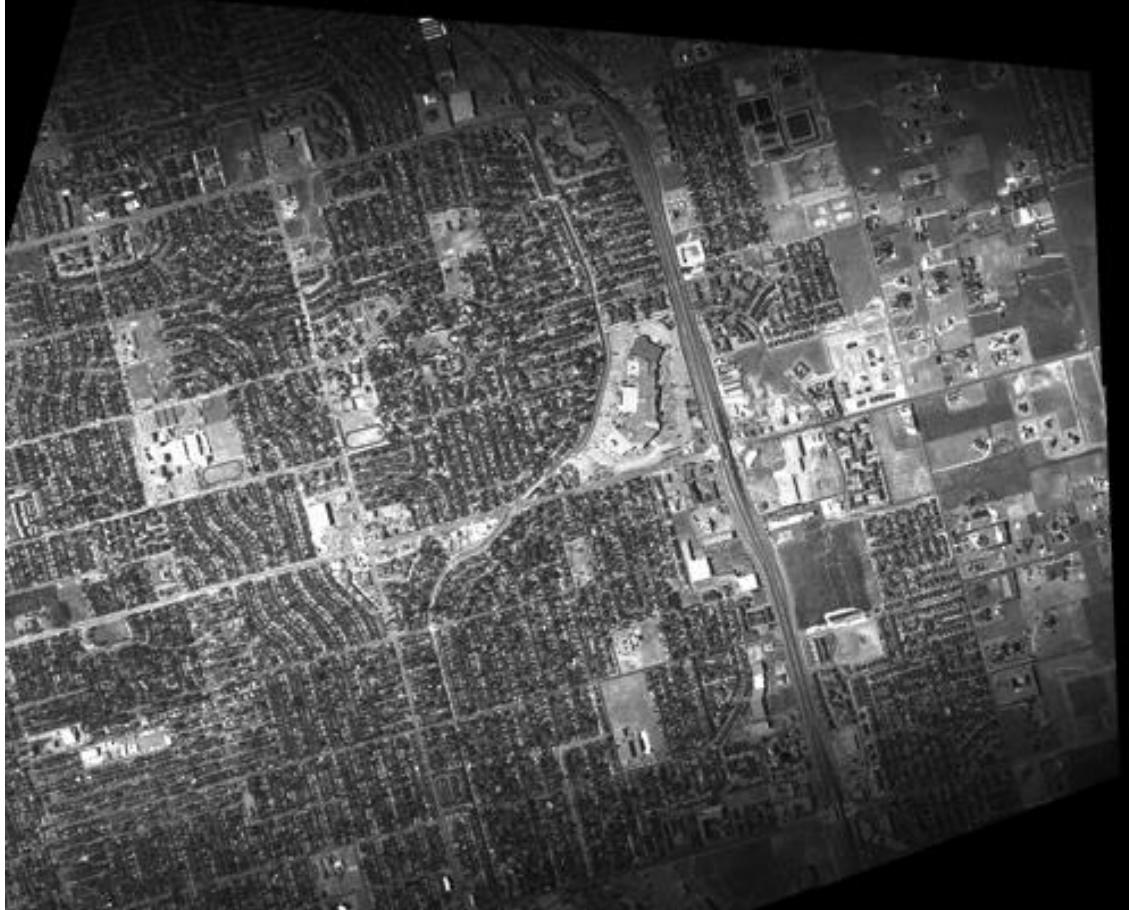


Apollo IMU

https://en.wikipedia.org/wiki/Inertial_measurement_unit

Computer Vision for Aerial City-Wide Persistent Surveillance

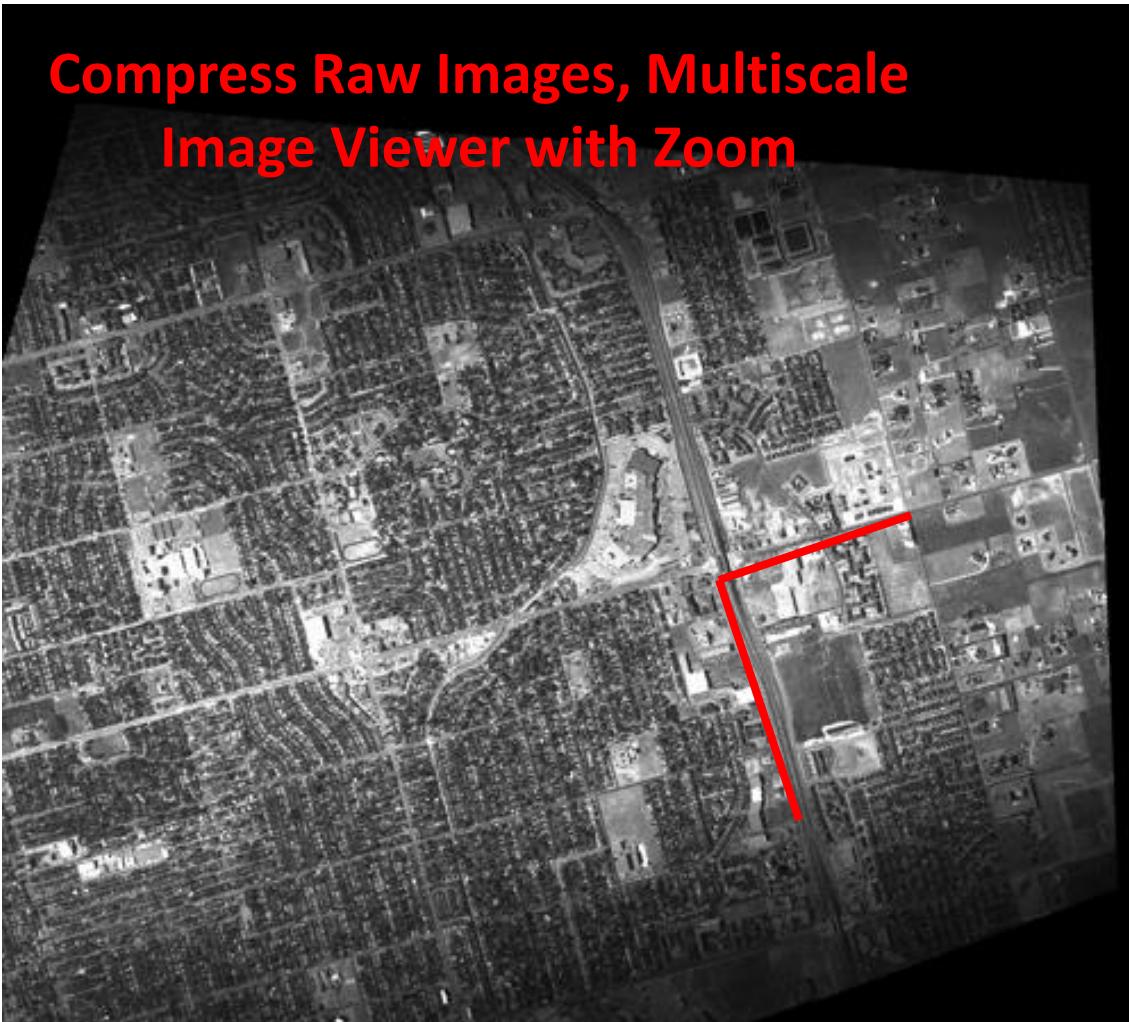
Register Rotated Imagery



Computer Vision for Aerial City-Wide Persistent Surveillance



Computer Vision for Aerial City-Wide Persistent Surveillance



Computer Vision for Multi-Projector Scalable Displays Calibrated By Multiple Cameras



Multi-Projector Scalable Displays



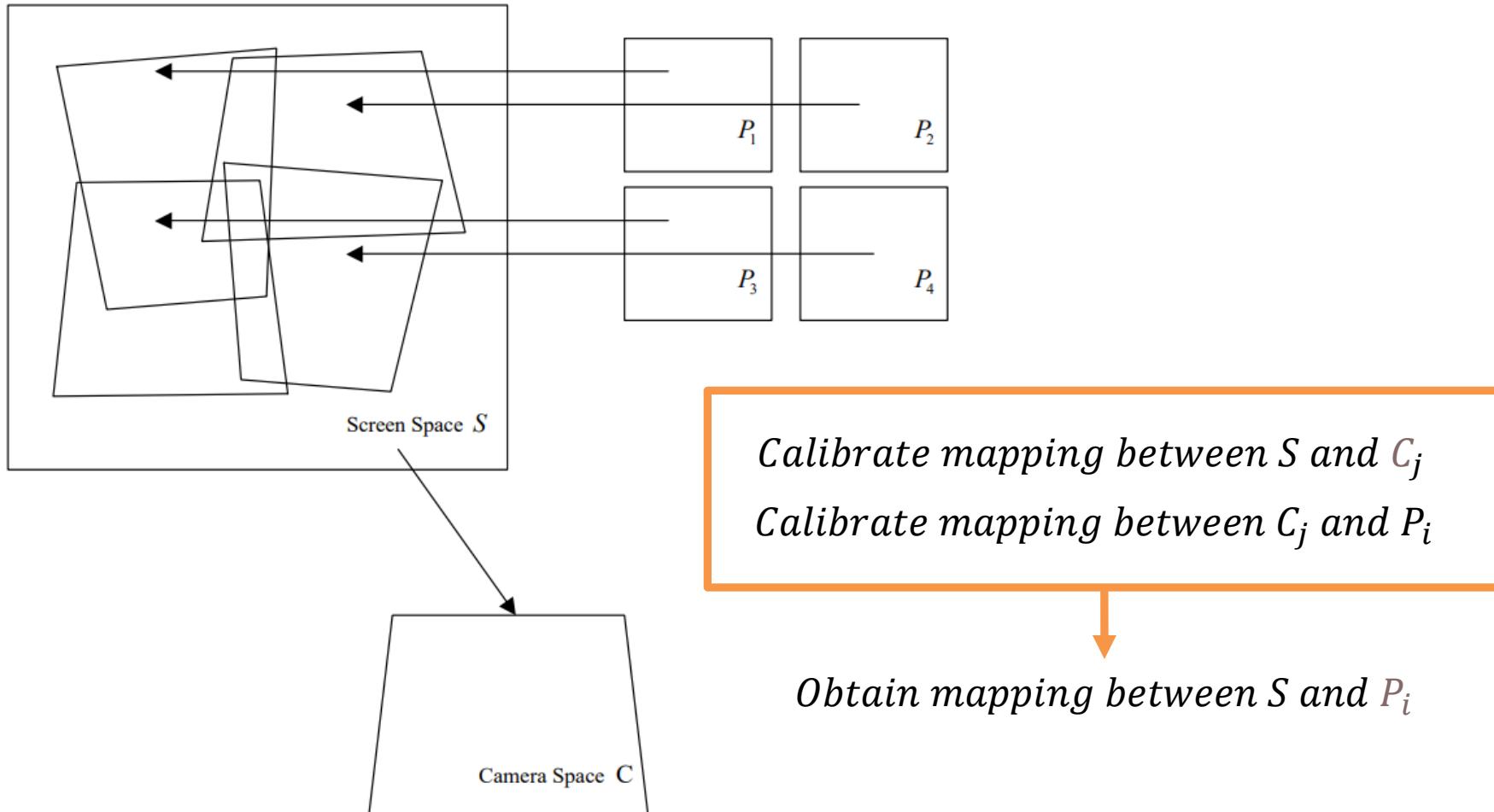
<https://www.scalabledisplay.com/gallery/>

Multiple Projectors Tiling the Screen



<https://www.scalabledisplay.com/gallery/>

Camera(s) and Projector(s) and Screen



Self-Calibrated Multi-Projector Displays

Wide Flat Screen



<https://www.scalabledisplay.com/gallery/>

Self-Calibrated Multi-Projector Displays

Cylinder Screen



<https://www.scalabledisplay.com/gallery/>

Self-Calibrated Multi-Projector Displays

Dome Screen



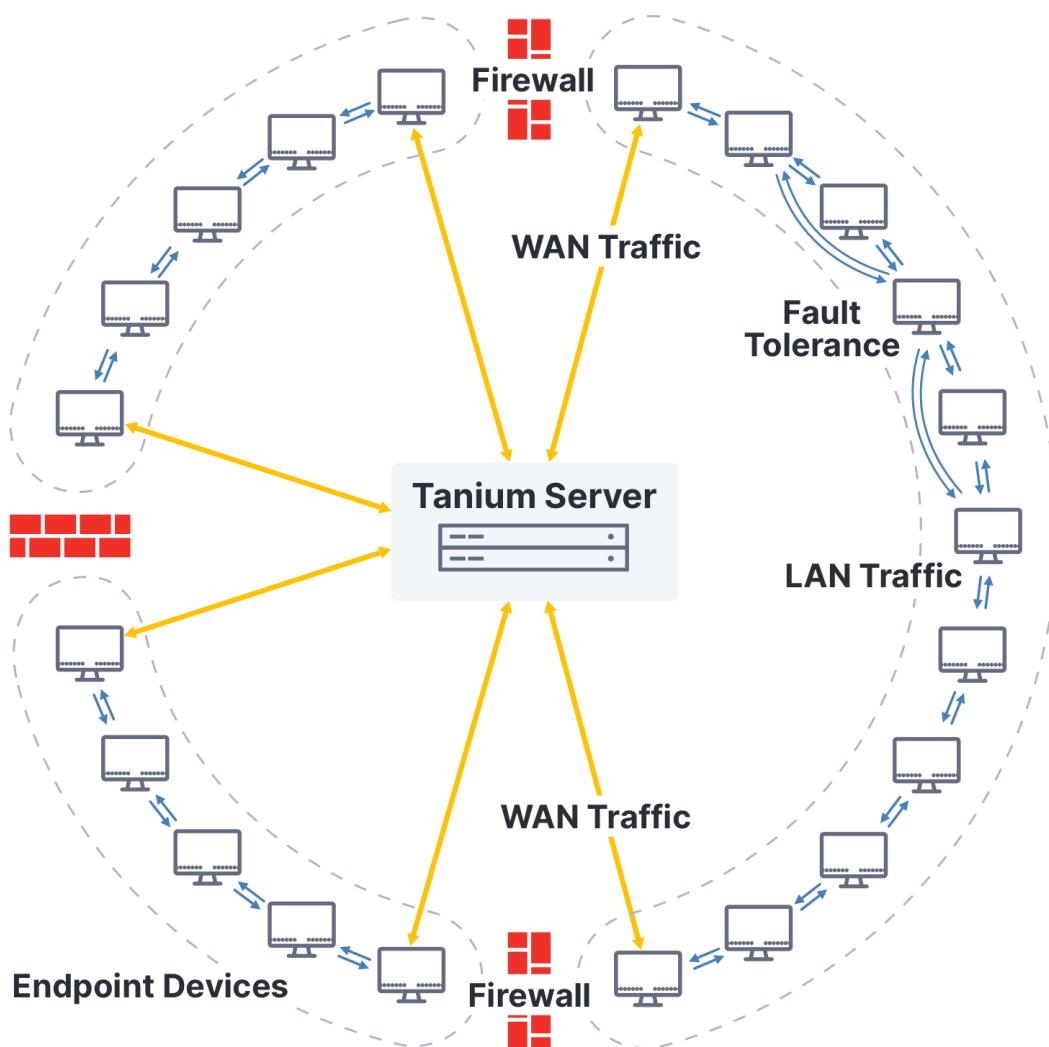
<https://www.scalabledisplay.com/gallery/>

Part III

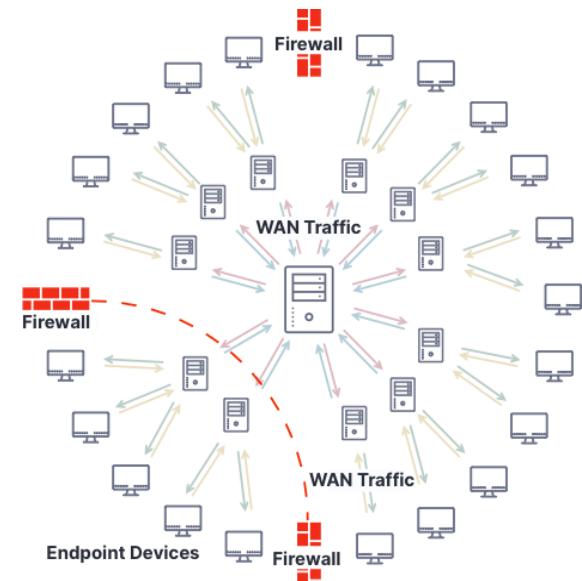
Federated Search Engines and Federated Learning

Tanium Server, Data Science	 TANIUM™	Industry
Federated Machine Learning	 TANIUM™	R&D
Federated Search Engine	VLDB-2021 	Industry, R&D

Tanium Server and Client Architecture



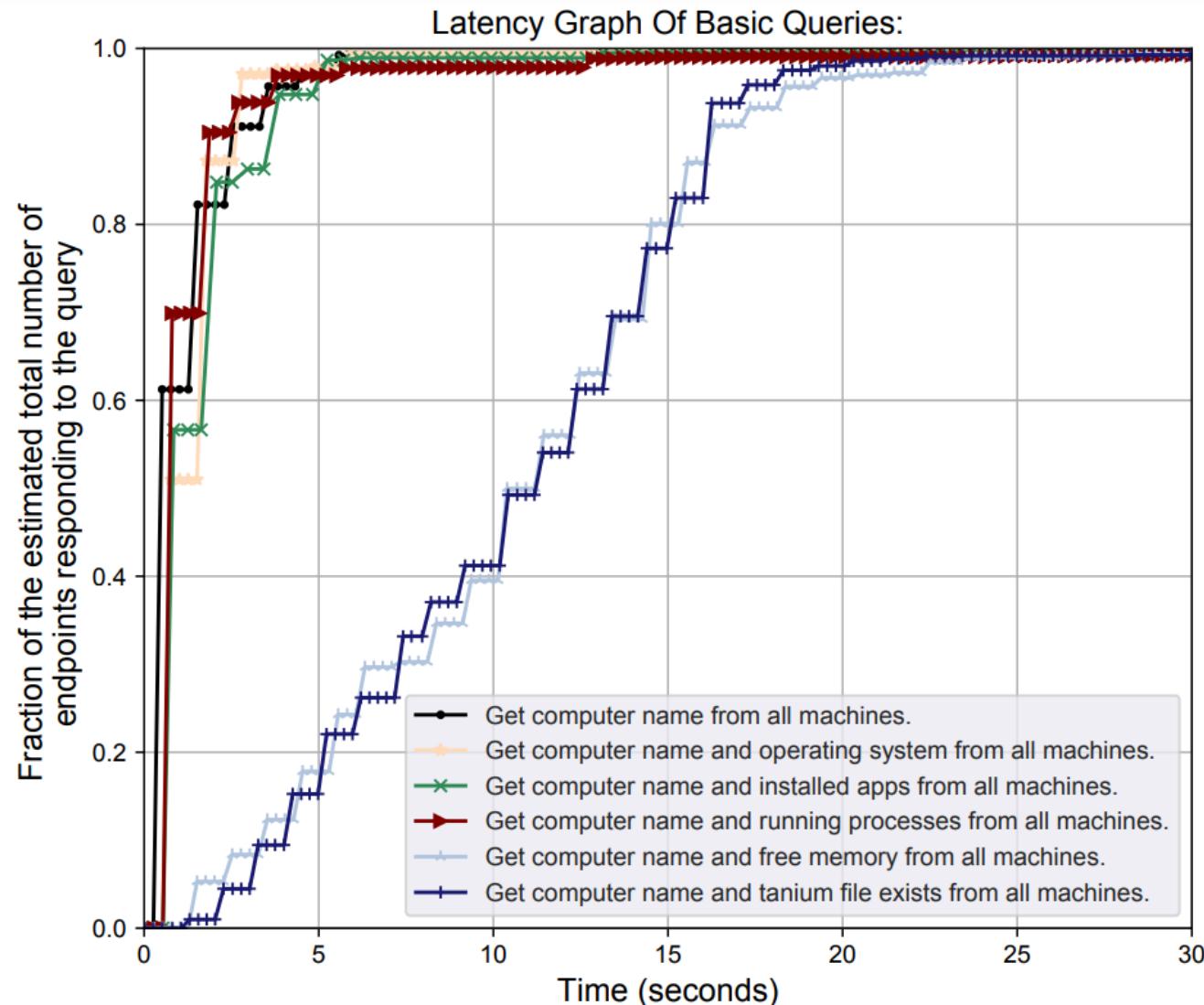
Tanium Linear Chain



Tree Topology

Other
Client-to-Cloud
Architectures

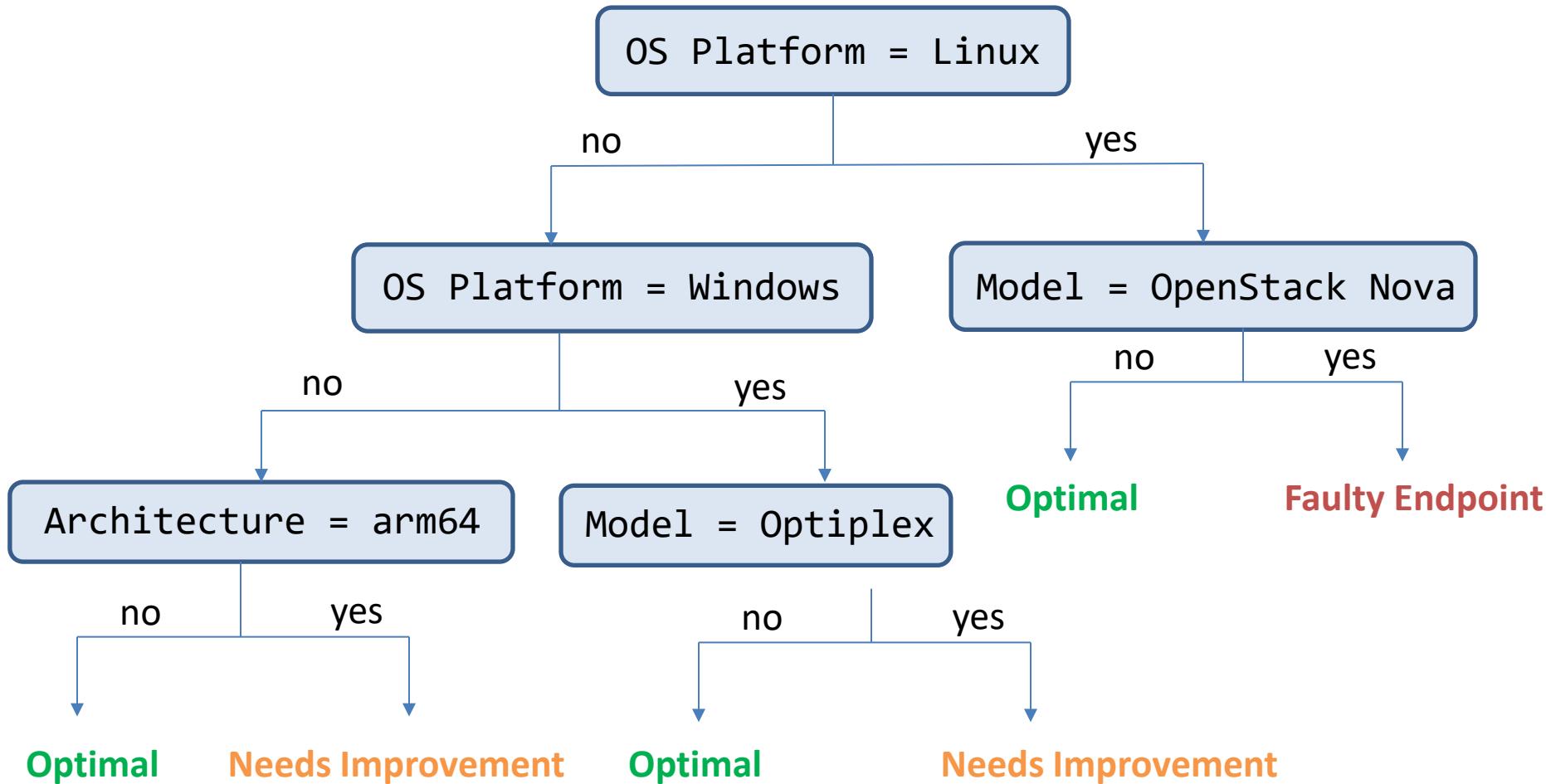
Ask The Tanium Server Questions



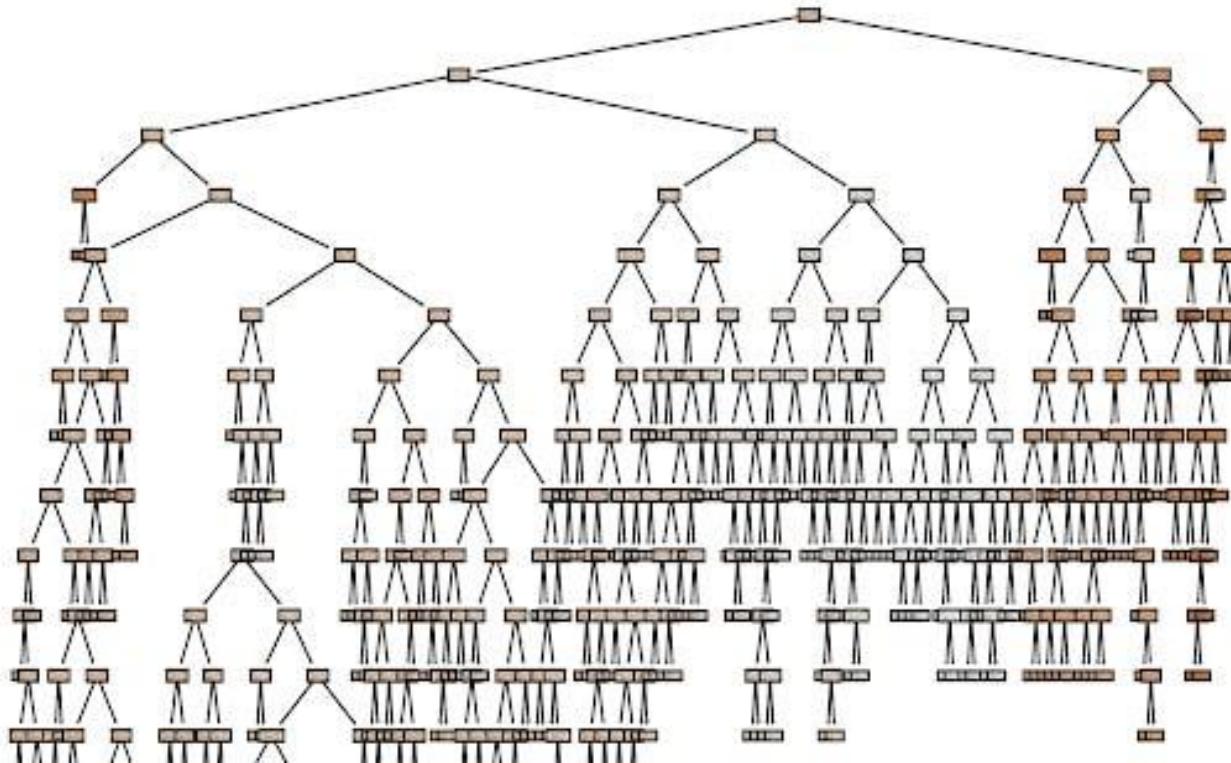
Get Data From 100,000+ Network Devices

Computer Name	Architecture	Average CPU Utilization Per 24h	OS Platform	Model	Performance Issues Per 24h
test-server	x86_64	12%	Linux	VMware Virtual	1
win-pc-24	x64-based PC	7%	Windows	VMware	2
client-local	x64-based PC	1%	Windows	Optiplex	4
matt-comp	x86_64	5%	Mac	MacBook Pro	0
mac-test	arm64	2%	Mac	Apple Virtual	3
nova-client	i686	24%	Linux	OpenStack Nova	8

Decision Tree Classifiers, Regressors



Decision Trees, Random Forests



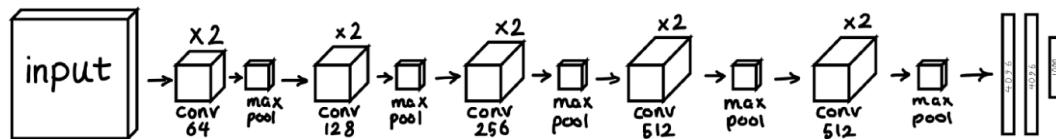
<https://cnvrg.io/decision-trees-python/>

Federated Machine Learning on Linear Communication Networks



Federated Machine Learning

OMITTED



Neural Network VGG13 Blog:

https://medium.com/@amir_hf8/implementing-vgg13-for-mnist-dataset-in-tensorflow-abc1460e2b93

VGG-13, 1.5M parameters

Alex Net, 62M parameters

VGG16 Net, 138M parameters

Llama-3, 70B parameters

GPT-3, 175B parameters

Claude 3.5 Sonnet, (180B? parameters)

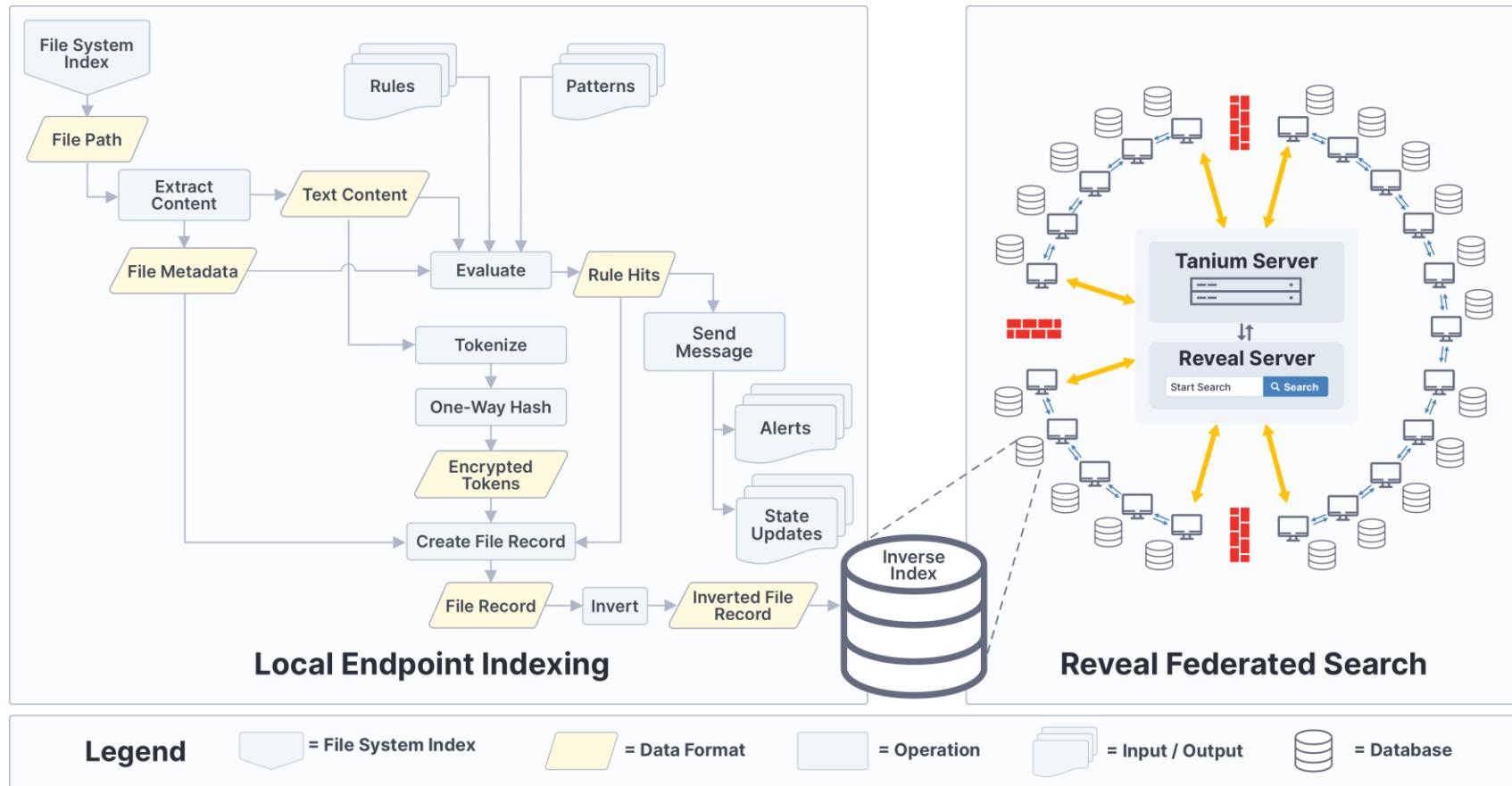
GPT-4, (1.76T? parameters)

Tanium Reveal: A Federated Search Engine for Querying Billions Of Files on Enterprise Networks

VLDB-2021



Federated Indexing and Search



Conceptual & Visual Design Credit: Christopher Poos

J. Stoddard, A. Mustafa, N. Goela, **Tanium Reveal: A Federated Search Engine for Querying Unstructured File Data on Large Enterprise Networks**, VLDB-2021, vol. 14, no. 12, pp. 3096-3109, Copenhagen, Denmark.

Federated Indexing and Search

STATISTICS	ENTERPRISE A	ENTERPRISE B
Reveal Endpoints	1273	4236
Files Indexed	427 Million	1.05 Billion
REVEAL INDEX SIZE	DISTRIBUTION A	DISTRIBUTION B
0-10MB	222 (17.4%)	42 (1.0%)
10-100MB	16 (1.3%)	47 (1.1%)
100-500MB	79 (6.2%)	53 (1.3%)
500MB-1GB	94 (7.4%)	300 (7.1%)
1-2GB	259 (20.3%)	1777 (41.9%)
2-5GB	603 (47.4%)	2017 (47.6%)
INCOMPLETE DATA		
With Dropped Files	3 (0.2%)	90 (2.1%)
Without Dropped Files	1270 (99.8%)	4146 (97.9%)

Federated Search – 3 Tiers



Search for the keyword “**Confidential**” across the enterprise

Federated Search – 3 Tiers



Search for the keyword “**Confidential**” across the enterprise

Filters				
Connection Status	Computer Name	Operating System	Hits ↓	Operation Status
<input type="checkbox"/>	<input checked="" type="radio"/> Disconnected	ap-Windows7-A.iw.apena.net	Windows 7 Professional	600 - 699 Ok
<input type="checkbox"/>	<input checked="" type="radio"/> Disconnected	M156-Ubuntu	Ubuntu 18.04.5 LTS	60 - 69 Ok
<input type="checkbox"/>	<input checked="" type="radio"/> Disconnected	5280.rjvaughn.lab	Oracle Linux Server release 8.1	20 - 29

Federated Search – 3 Tiers



Search for the keyword “**Confidential**” across the enterprise

Filters				
▶ II				
Connection Status	Computer Name	Operating System	Hits ↓	Operation Status
<input type="checkbox"/>	Disconnected	ap-Windows7-A.iw.apena.net	Windows 7 Professional	600 - 699 Ok
<input type="checkbox"/>	Disconnected	M156-Ubuntu	Ubuntu 18.04.5 LTS	60 - 69 Ok
<input type="checkbox"/>	Disconnected	5280.rjvaughn.lab	Oracle Linux Server release 8.1	20 - 29

Federated Search – 3 Tiers



Search for the keyword “**Confidential**” across the enterprise

Filters				
Connection Status Computer Name Operating System Hits ↓ Operation Status				
<input type="checkbox"/>	Disconnected	ap-Windows7-A.iw.apena.net	Windows 7 Professional	600 - 699 Ok
<input type="checkbox"/>	Disconnected	M156-Ubuntu	Ubuntu 18.04.5 LTS	60 - 69 Ok
<input type="checkbox"/>	Disconnected	5280.rjvaughn.lab	Oracle Linux Server release 8.1	20 - 29 Ok

A blue arrow points from the bottom of the first table to the second table.

Filename		Hits ↓	Directory
<input type="checkbox"/>	authorization_code.py	2	/snap/core/1018n3/dist-packages/oauthlib/oauth2/rfc6749/g
<input type="checkbox"/>	resource_owner_password_credentials.py	2	/snap/core/1018n3/dist-packages/oauthlib/oauth2/rfc6749/g
<input type="checkbox"/>	authorization_code.py	2	/snap/core/1012n3/dist-packages/oauthlib/oauth2/rfc6749/g
<input type="checkbox"/>	resource_owner_password_credentials.py	2	/snap/core/1012n3/dist-packages/oauthlib/oauth2/rfc6749/g

Federated Search – 3 Tiers



Search for the keyword “**Confidential**” across the enterprise

Filters				
Connection Status Computer Name Operating System Hits ↓ Operation Status				
<input type="checkbox"/>	Disconnected	ap-Windows7-A.iw.apena.net	Windows 7 Professional	600 - 699 Ok
<input type="checkbox"/>	Disconnected	M156-Ubuntu	Ubuntu 18.04.5 LTS	60 - 69 Ok
<input type="checkbox"/>	Disconnected	5280.rjvaughn.lab	Oracle Linux Server release 8.1	20 - 29 Ok

Filename		Hits ↓	Directory
<input type="checkbox"/>	authorization_code.py	2	/snap/core/1018n3/dist-packages/oauthlib/oauth2/rfc6749/g
<input type="checkbox"/>	resource_owner_password_credentials.py	2	/snap/core/1018n3/dist-packages/oauthlib/oauth2/rfc6749/g
<input type="checkbox"/>	authorization_code.py	2	/snap/core/1012n3/dist-packages/oauthlib/oauth2/rfc6749/g
<input type="checkbox"/>	resource_owner_password_credentials.py	2	/snap/core/1012n3/dist-packages/oauthlib/oauth2/rfc6749/g

Federated Search – 3 Tiers



Search for the keyword “**Confidential**” across the enterprise

Filters				
Connection Status Computer Name Operating System Hits ↓ Operation Status				
<input type="checkbox"/>	Disconnected	ap-Windows7-A.iw.apena.net	Windows 7 Professional	600 - 699 Ok
<input type="checkbox"/>	Disconnected	M156-Ubuntu	Ubuntu 18.04.5 LTS	60 - 69 Ok
<input type="checkbox"/>	Disconnected	5280.rjvaughn.lab	Oracle Linux Server release 8.1	20 - 29 Ok

Filename		Hits ↓	Directory
<input type="checkbox"/>	authorization_code.py	2	/snap/core/1018n3/dist-packages/oauthlib/oauth2/rfc6749/g
<input type="checkbox"/>	resource_owner_password_credentials.py	2	/snap/core/1018n3/dist-packages/oauthlib/oauth2/rfc6749/g
<input type="checkbox"/>	authorization_code.py	2	/snap/core/1012n3/dist-packages/oauthlib/oauth2/rfc6749/g
<input type="checkbox"/>	resource_owner_password_credentials.py	2	/snap/core/1012n3/dist-packages/oauthlib/oauth2/rfc6749/g

```Authorization Code Grant`\_

The authorization code grant type is used to obtain both access tokens and refresh tokens and is optimized for **confidential** clients. Since this is a redirection-based flow, the client must be capable of interacting with the resource owner's user-agent (typically a

\*Novelty

## Tanium vs. Google Search Engines

### Novel System Parameters of Tanium Reveal

SEARCH ENGINE	CENTRALIZED [8]	FEDERATED
Data Volume	>100B pages [24] World Wide Web	>1B files/enterprise >100k files/endpoint >10k endpoints
<b>INDEXING</b>		
Index Location	cloud data center	endpoint devices
Index Size	>100 PB [24]	<10 GB/endpoint
Index Compute	cloud compute	<5% CPU/endpoint
Network Cost	linear in data	$O(1)$ in data
Index Refresh	3-28 days [59]	<12 hours
<b>SEARCH</b>		
Query Latency	<0.1s [80]	<60s
Query Frequency	>80k queries/s [36]	<100 queries/s
<b>DATA</b>		
Data At Edge	no	yes
Sensitive Data	no	yes

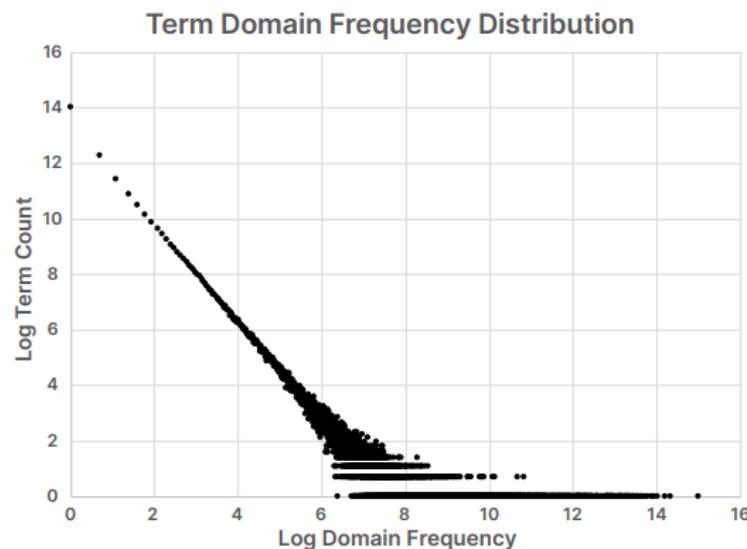
\*Novelty

## Tanium Reveal: Novel Indexing at Petabyte Scale

Table 5: Wikipedia Index Statistics

DESCRIPTION	INDEXING THE WIKIPEDIA CORPUS
Corpus Contents	5,325,951 Files
Corpus Size	122 GB (131,187,937,280 bytes)
Text Extracted	18 GB (19,733,555,872 bytes)
Files Indexed	5,305,603 (20,348 failed)
Index Size	46 GB (50,144,695,764 bytes)
Unique Tokens	1,849,541 (615,585 in multiple files)
Indexing time	9 Hours (31,596 Sec) @ 10% CPU

Indexing Wikipedia (122-GB)



Indexing Enterprise Data: Petabyte Scale  
(20-GB per endpoint, 100k endpoints = 2-PB)

Figure 6: The number of unique tokens (y-axis) appearing at each domain frequency (x-axis) within the Wikipedia corpus, logarithmically scaled.

