FIAS Frankfurt Institute for Advanced Studies

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

www.goethe-universitaet.de

Systems Engineering Meets Life Sciences: Review

Prepared by:
Prof. Dr. Visvanathan Ramesh

Summary of Lectures

System engineering examples

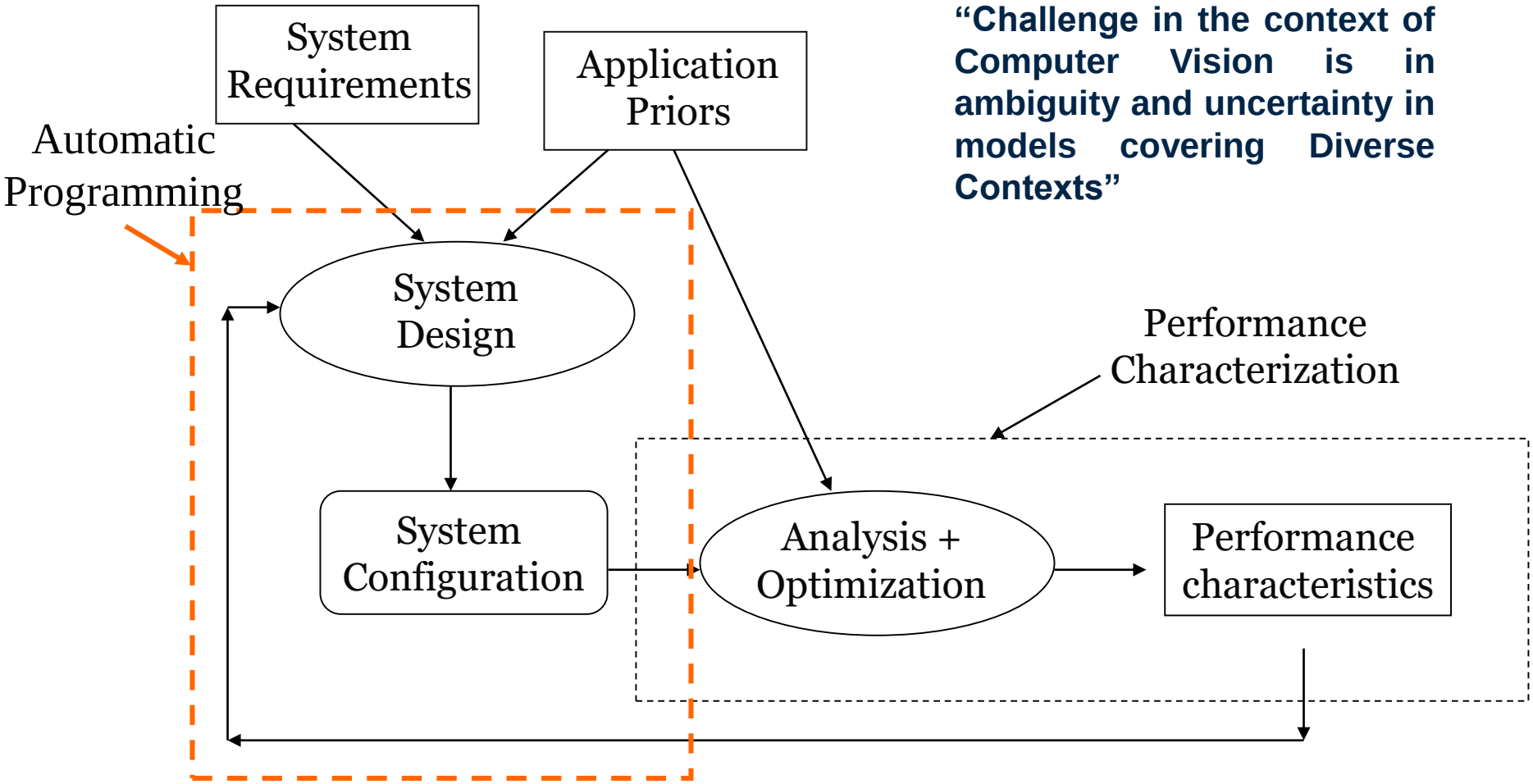
- Greiffenhagen et al (2001)
- S. Veerasavarappu et al (2013-17) Simulation for vision
- R. Hota et al (2013-17), Weis et al (2015-17)

Overall theme

(Context,Task,Performance) → Hw plus Sw configuration (hw + programs plus parameters)

- Context, Task, Performance
  - What is context -- Derek hoeim's Book (2015)
  - Task - estimation of world state (or parts of it)
  - Performance - bias, variance , accuracy vs speed tradeoff
- What is a Program (Inference Engine)?
  - Program - filters and combinations (feedforward, deep, feedback and recurrent) (ML Literature, Bio-inspired vision literature 2016)
  - Program Design – Model based vs Data Driven, or Hybrid combinations
  - Model based design - (Mann, 1996)
    - Graphical model illustration using vscp
    - Inference Aggregation Indexing , prediction, verify loops , Hierarchical
- What about Performance? (Ramesh, 1995)

Systems Design & Analysis Cycle



"Challenge in the context of Computer Vision is in ambiguity and uncertainty in models covering Diverse Contexts"

Requirements Specification for Real-time Vision Systems

FIAS Frankfurt Institute for Advanced Studies

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Input Space specification:

- Object-oriented Graphical Models describing generative models for video data given scene variables
- Scene variables include:
  - Scene Geometry (static geometry), Material distribution, Environmental Conditions (e.g. weather, indoor, outdoor), Object types in the scene, their shape, dynamics, Illumination distribution (e.g. source positions, dynamics), Camera (Sensor) positions, orientations in the world, projection geometry, photometric model

Task Specification:

- Desired subset of scene parameters to be estimated from video (for example):
  - Counts of object
  - Object types, Object tracks, Object geometry, Object behavior
  - Analysis of Groups of objects
  - Illumination/weather state

Performance Requirements:

- For each task: probability of error (e.g. p_miss, p_false in two class situations)
- Accuracy in Parameter estimates (tolerances)
- Graceful degradation, Self-Diagnosis
- Computational speed
- Time delay to respond (i.e for computation of results), etc.

Desired Properties of Vision System Designs

FIAS Frankfurt Institute for Advanced Studies

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

- **Modularity in Specifications:**
  - Nested model spaces to allow for various degrees of approximations in the model space
- **Scalability of Design Solutions:**
  - Ability to derive families of solutions where the complexity of system is scaled according to complexity of tasks, input space approximations.
- **Quantifiability:**
  - Ability to provide quantitative performance models of system designed as a function of Graphical Model parameters and tuning parameters/constants of system.
- **Computational Complexity tradeoff vs Accuracy:**
  - Ability to quantify computational complexity of system as function of OODBN parameters.
  - Use this quantification to provide tradeoffs (e.g.) Reduce accuracy for reducing computation.
- **Modular Extensibility:**
  - Design should allow for modular extensibility when input spaces in one application differ from another in a minor way.
- **Mapping to Hardware:**
  - Design should allow ease of mapping to target hw. (could address this as a separate phase. (i.e). Construct designs for general purpose architectures and then have a systematic approach to translation of design to hw.

Paradigms for Design

- Model Based Design
  - Generative Models – i.e. Probabilistic Graphical Models (Interpretation is estimation of world state given observations. Generative model uses a likelihood model for sensor observations (physics-based) and Prior model.)
- Data Driven Machine Learning
  - Neural Networks
  - Boosting, Support Vector Machines, etc.
- Hybrid designs (combination)

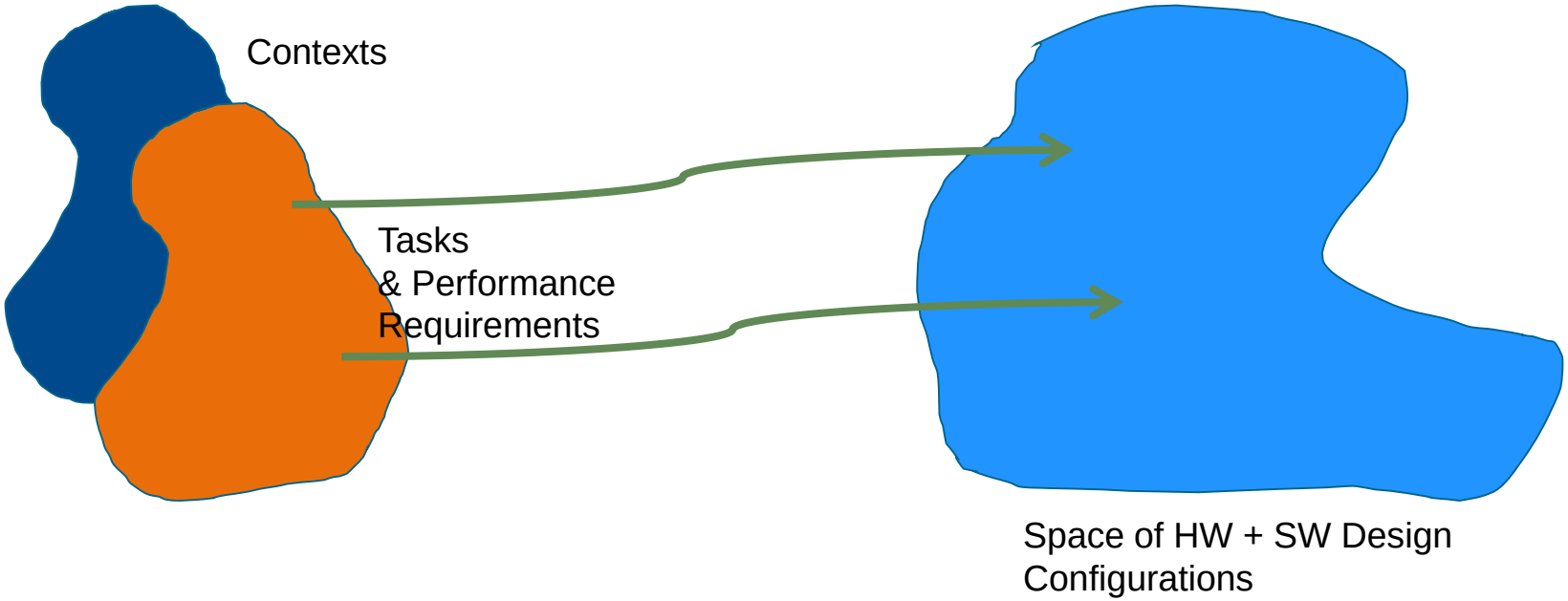
Systems Engineering: Key Ideas

FIAS Frankfurt Institute for Advanced Studies

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

- Formalize domain (i.e. generative) models for application contexts
- Formalize system task requirement specification
- Translate requirements to formal generative models
- Link generative models to approximate inference engines (i.e. module and system implementations)
- Performance characterization of design (white box analysis)
- Model Validation and Iteration of Design (comparison of empirical and theoretical predictions and model/design improvement)

Key Insight: Learning of (C, T, P) → Program mappings



"(Contexts, Task and Performance Requirements) → to (System Designs)"
Extension to new design settings – via re-use of context elements and identification of gaps in models

Methodology Summary

User View Of System

Application Domains:

- Automated Video Surveillance for Safety and Security
- Vision for Autonomous driving – comfort and safety

Task Specification:

e.g. A situational aware system that: parses video from indoor and outdoor environments and generates descriptions of objects, events and is able to alert to a user when event descriptions match user specified queries.

Performance Specs:

(Functional as well as non-functional):
For each task: probability of error (e.g. probability of miss detection, probability of false alarm in two class situations)

- Accuracy in Parameter estimates (tolerances)
- Graceful degradation, Self-Diagnosis
- Cost, Computational speed
- Time delay to response, etc.

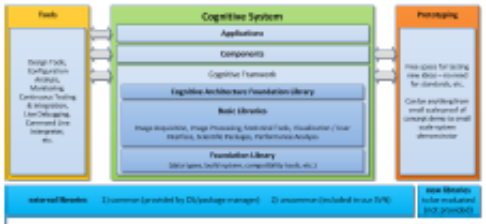
Modeling View of System

Formal Graphical Models of Context (of various degrees of approximation) along with Physics based Sensor models are linked to systems level analysis of sequential estimators.

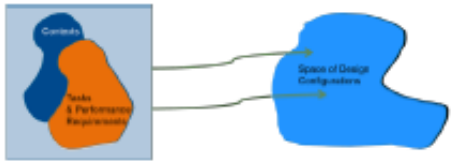
Overview: A vision system is complex and composed of multiple stages of execution and is a non-linear estimator from Sensed Images → Semantics. Choice of estimator depends on application requirements and graphical models and involves approximations. Core ideas use in the methodology include:

- Statistical methods for describing population of inputs, quantifying performance and limitations, and to perform inference.
- Use of Probabilistic Contextual Models - Represent application constraints in terms of: priors on camera parameters and projection model, object parameters and dynamics, priors on illumination.
- Develop likelihood models L(features|image) taking into account sensor and feedforward computational process models correctly. Perform Bayesian inference by combining likelihood model and priors.
- Understanding of Vision application space → likelihood model choices and prior choices leads to the building of systems in a rapid and cost effective way.

Implementation View of System

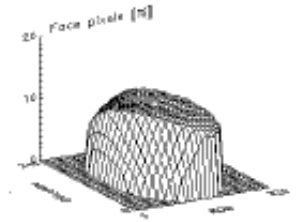


Cognitive Vision Platform for Rapid Prototyping & Validation



"Explore Model Space for Contexts, Task and Performance Requirements, Systematically Map them to Design configurations, Prototype Designs, Validate, Iterate" as in 'Design Thinking'. Extension to new design settings – via re-use of context elements and identification of gaps in models."

Evaluation & Validation of System



System Performance Model (Example)

Validation. First two lines show the predicted (hat), and experimental (tilde) variances for the tilt angle θ. Next two lines correspond to pan angle δ.

$\times 10^{-3}$	P1	P2	P3	P4	P5	P6	P7	P8
$\hat{\sigma}^2_{\tan\theta}$	2.10	2.12	1.57	1.40	1.35	1.31	1.31	1.32
$\tilde{\sigma}^2_{\tan\theta}$	2.05	2.04	1.60	1.34	1.36	1.32	1.40	1.31
$\hat{\sigma}^2_{\sin\delta}$	28.9	26.1	21.3	17.9	15.3	15.2	18.4	20.1
$\tilde{\sigma}^2_{\sin\delta}$	25.9	24.1	19.5	15.1	14.9	15.0	18.1	19.3

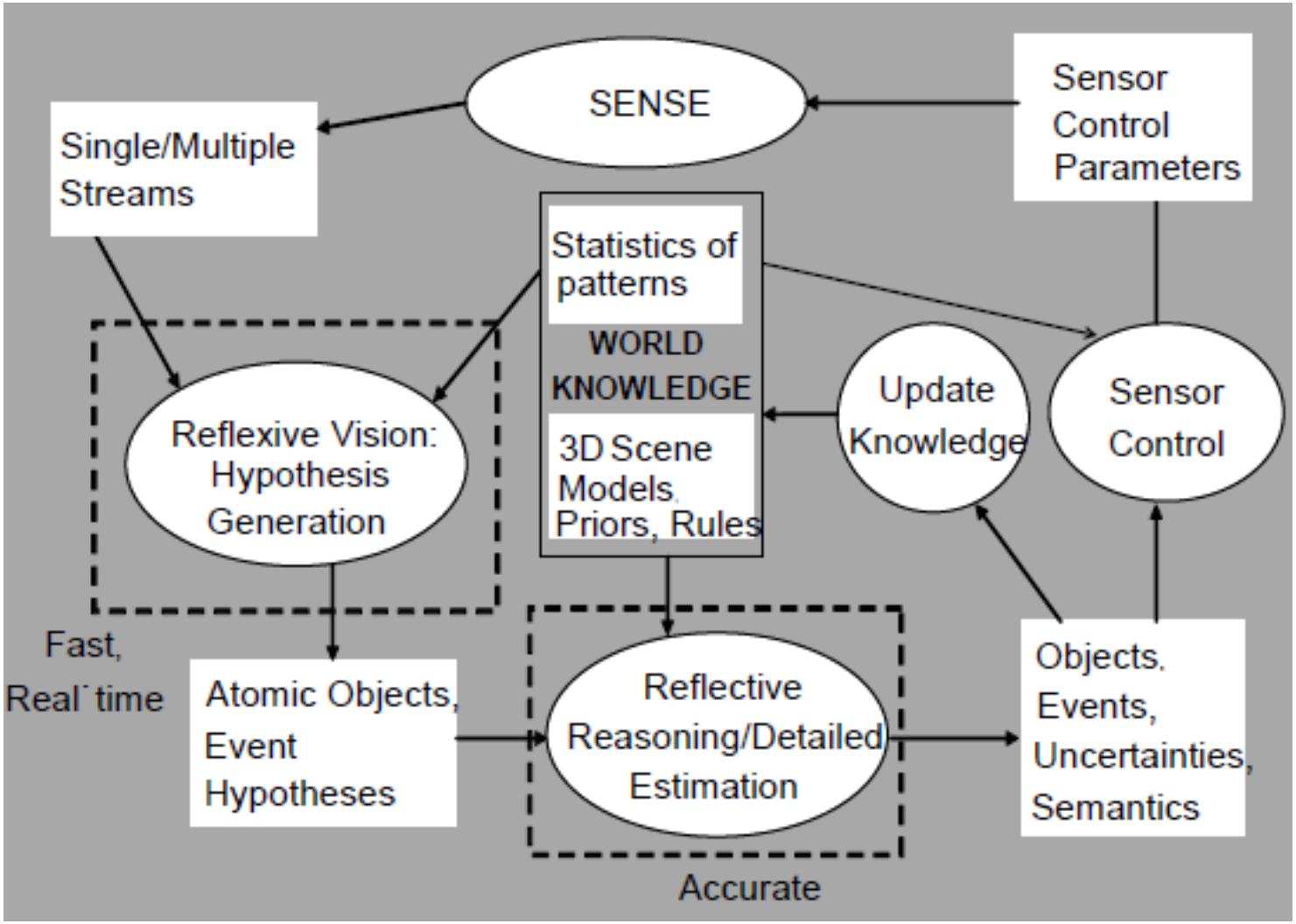
Model Validation (Theory vs Experiment)

Large scale Field Trials & Validation

"Visual Cognition is 'quasi-invariant Indexing' followed by detailed estimation (or deliberation, iteration) – component level research is quite mature, open research is on systems questions involving Cognitive Vision Platform with Continuous Learning and Self-Diagnostics".

Essence of Overall Design Framework: {Application contexts} x {sensor types + configurations} x {questions posed} x {perf specs/requirements} ----> {specific hypotheses generators} + {reasoning / optimization engine}

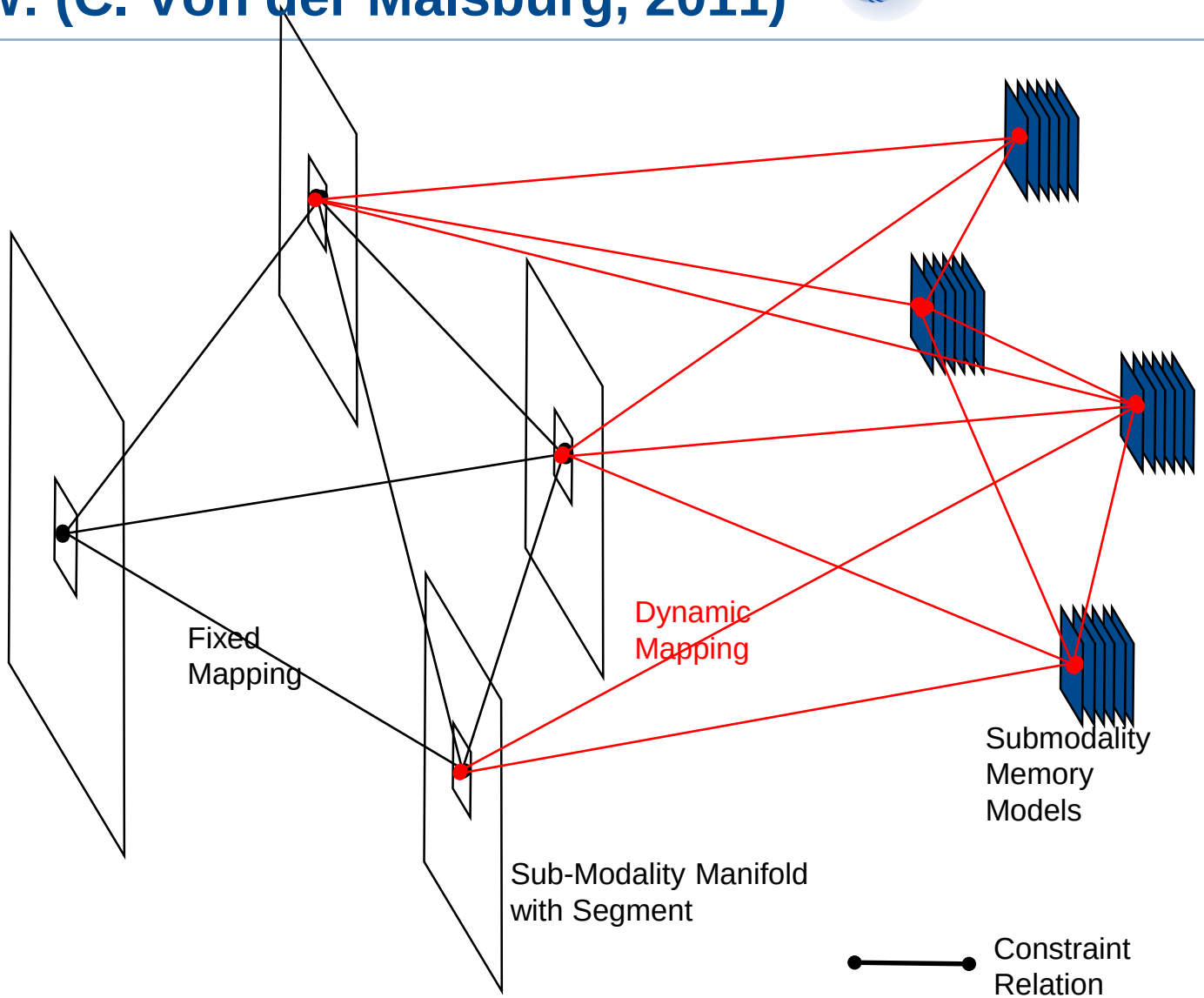
Visual Cognition: Hierarchical Indexing + Iterative Estimation



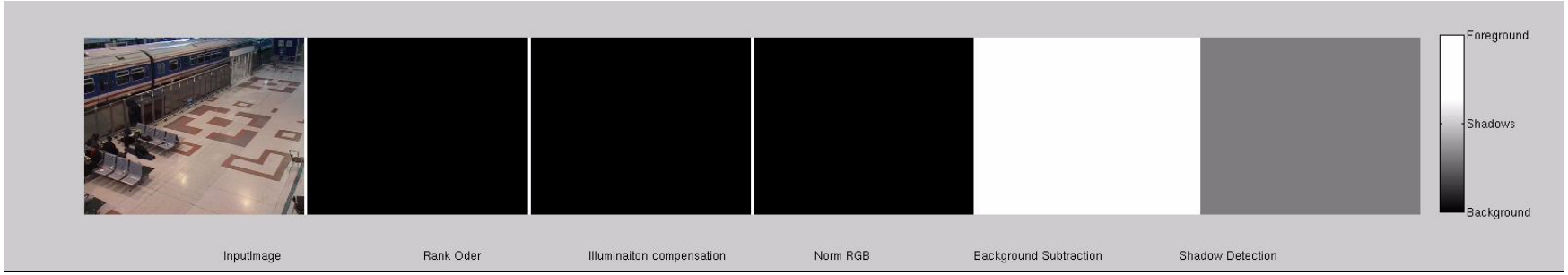
Modules
Sense
Hypotheses Generation
Reasoning/ Optimization
Fusion Strategy
Online Learning
Control & Action
Memory Representation

(VR)

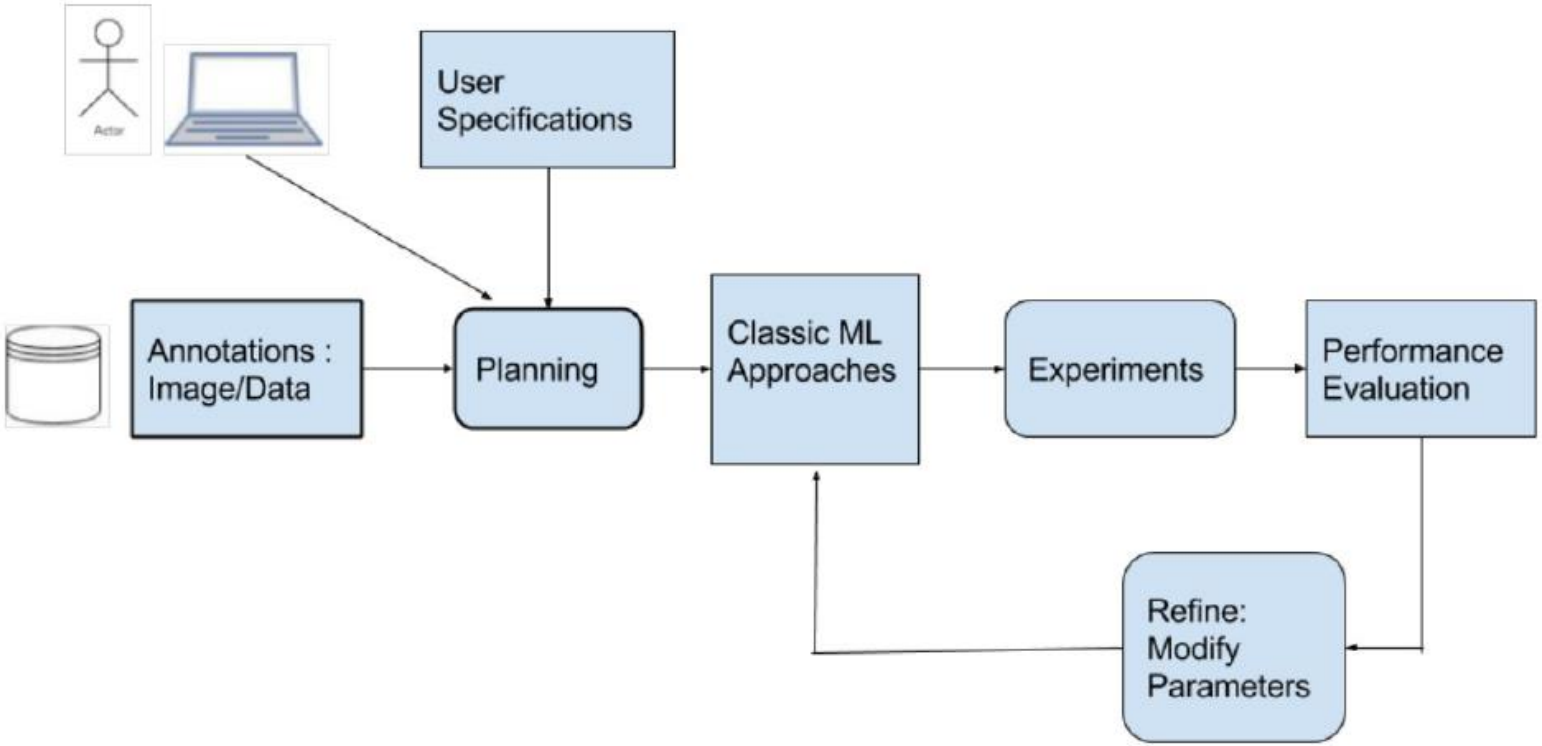
Computational Neuroscientist's View: (C. Von der Malsburg, 2011)



Demo Video Illustrating Decomposition



System Design Process



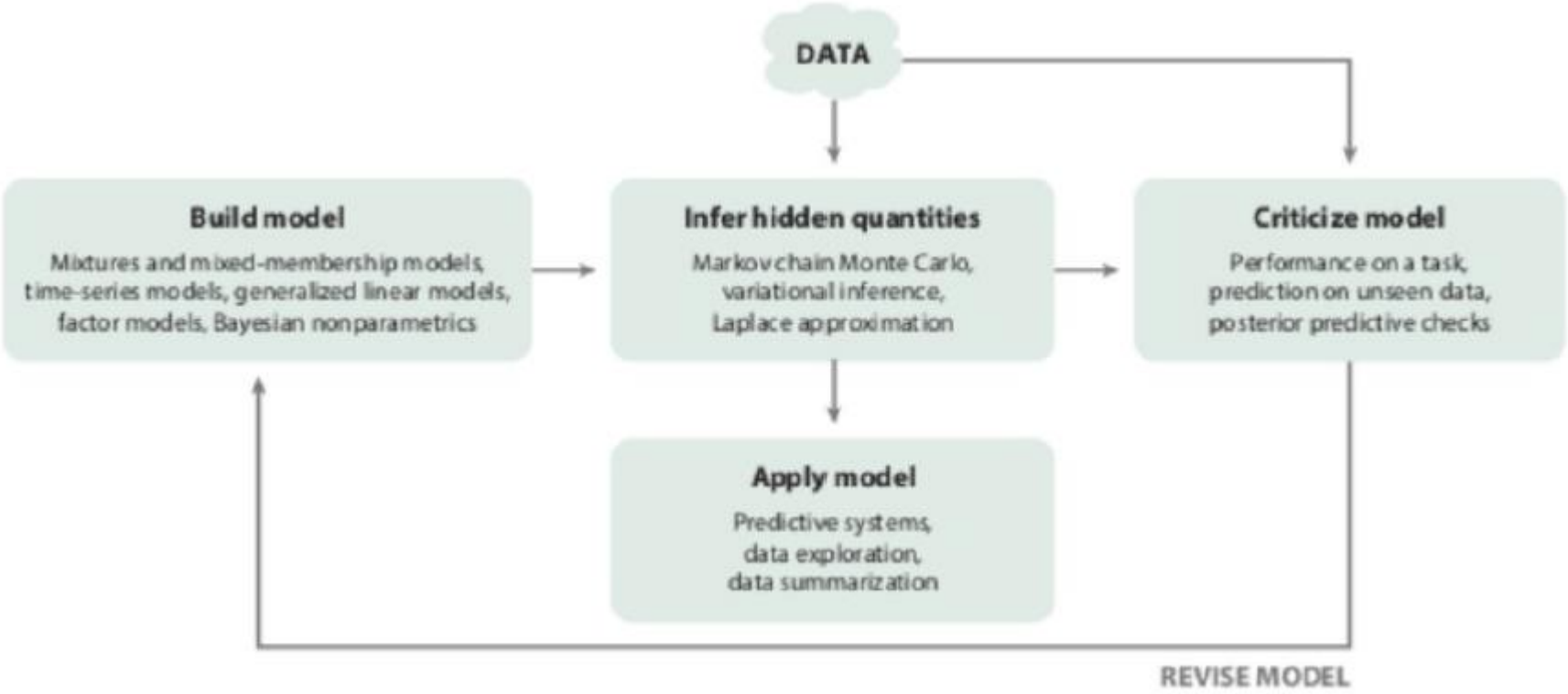
Design Work flow – From Skeleton Designs to performance evaluation

Solution Approaches

- Model Based Design
- Data Driven Design
- Hybrid Approach : Considering both model and data driven designs

Solution Approaches

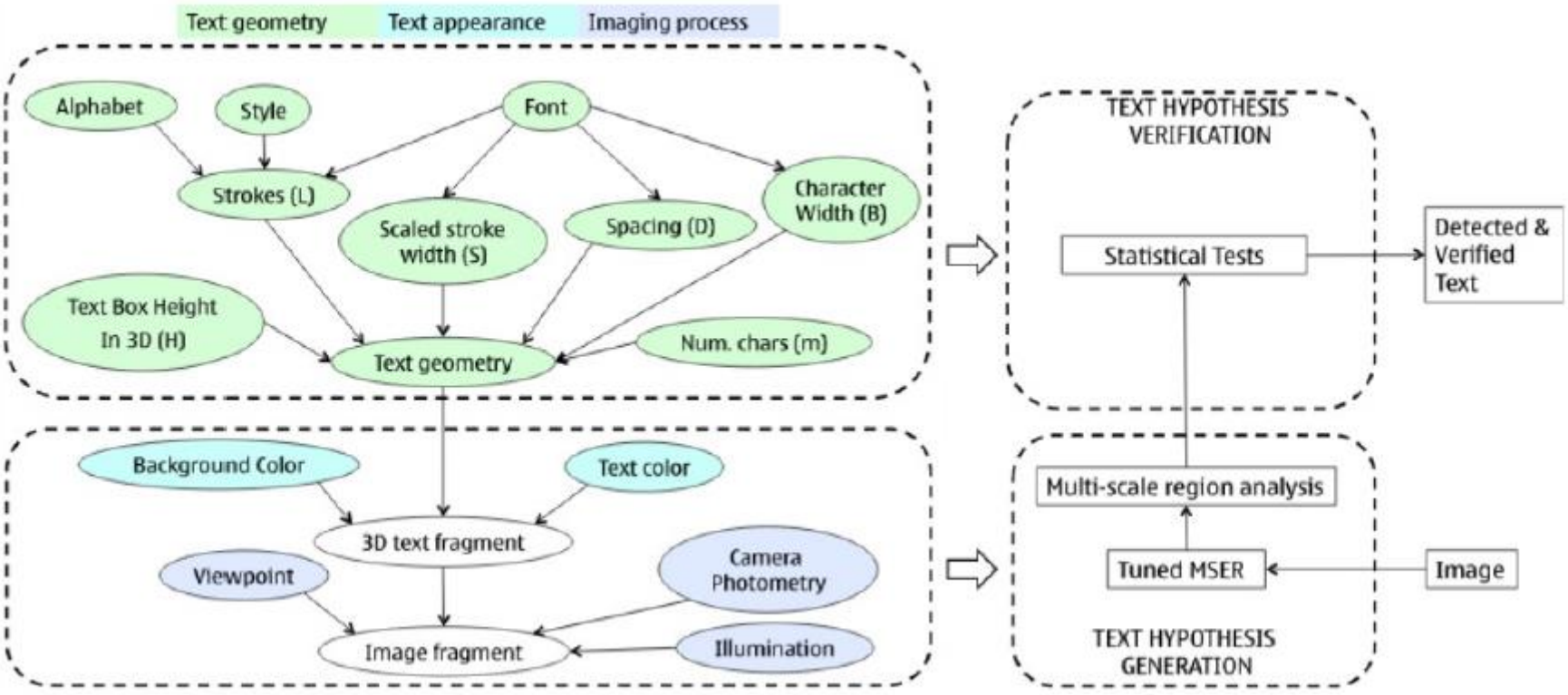
1. Model Based Design



General setup of Model based methods.
Image Source: [6], Blei (2015)

Solution Approaches

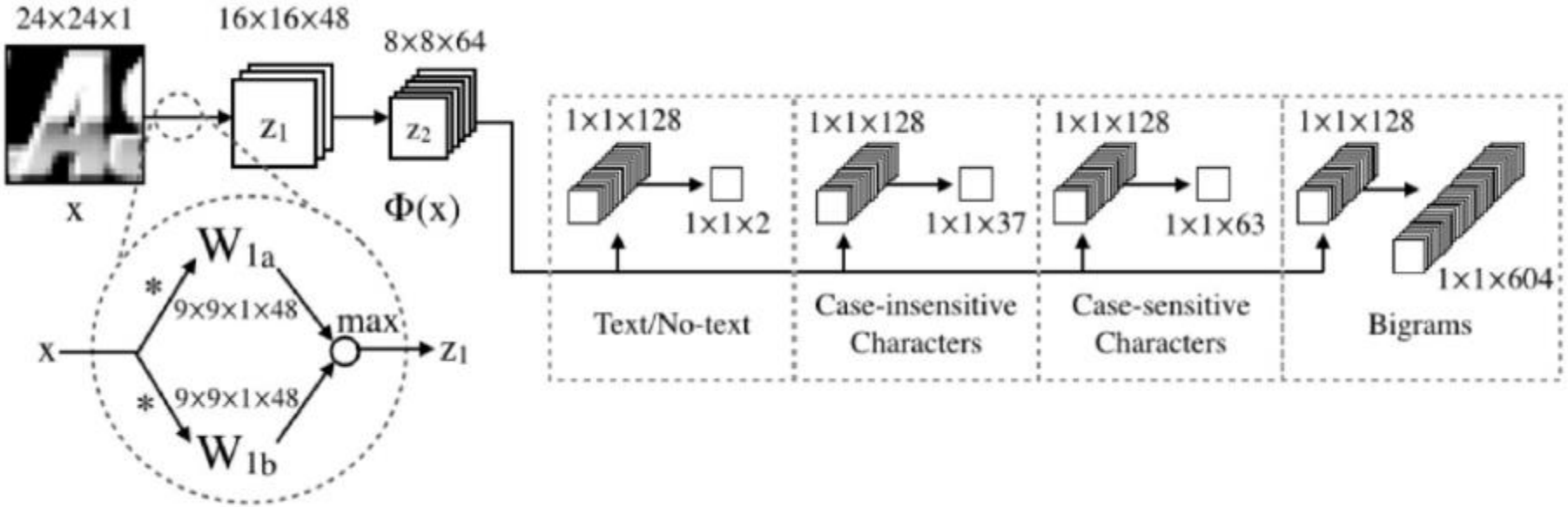
Classic Example for Model Based Design



Left: Bayesian Network for Text Appearance in an Image.
System Design (See [3])

Solution Approaches

2. Data Driven Design



Convolutional Neural Networks. The method uses four CNNs. These share the first two layers, computing "generic" character features and terminate in layers specialized into text/no-text classification, case-insensitive and case-sensitive character classification, and bigram classification. Each connection between feature maps consists of convolutions with maxout groups. Figure and caption from [7]

Solution Approach: Hybrid

- Combine strengths of model-based thinking as well as data driven machine learning.
- Several feature maps are extracted based on several feature extraction kernels.
- This is followed by a deep neural network architecture or any data driven architecture for the purpose of classification and recognition.

Contextual Models: Simulation from Generative Models

- Camera geometry -- projection model (orthographic, perspective), camera blur, lens distortion, intrinsic parameters, extrinsic parameters.
- Camera - gray level transformation model of camera pipeline
- Shape representation (surface/contour, volume)
- Material property (brdf)
- Appearance (texture map) dictionary
- Graphics pipeline parameters

Pattern Models

- Image models for textures
  - sparsity based
  - MRF model Pipelines
- Texture classification
  - Hand engineered features plus ML - deep learning
- Texture synthesis
  - Deep learning
  - Exemplar based with smoothness constraint (efros)
  - Bio-inspired (wavelets plus correlation, constrained sampling) - mrf models

Spatial Statistics - Simulation

Simulation Models –

- spatial point processes
  - poisson point process
    - in homogenous process
    - cluster processes (cox, matern hard core, etc)
    - Boolean germ-grain models
    - dead leaves model
- Simulation apparoach
  - rejection sampling
  - Markov Chain Monte Carlo
  - CFTP - Coupling from the past
  - Representations: Sparsity based reps, Texture models using MRF's

- Open questions:
  - Realism of models ?
  - Model validation against real data

Simulation for Systems Design, Analysis and Evaluation

FIAS Frankfurt Institute for Advanced Studies

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

- **Groundtruth collection seems to be an obstacle for Supervised learning based vision systems.**

- **Major advances in Computer Graphics (CG) field has spurred a renewed recent interest to utilize CG for CV.**

(a) 2001 (b) 2003 (c) 2005

(d) 2006 (e) 2013 (f) 2015

Figure 1-1: Evolution of Graphics in Video games from 2001 to 2016

Rendered Data - various scene conditions

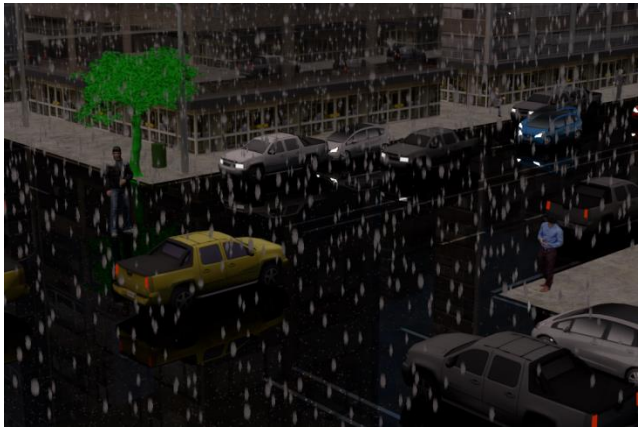
Lambertian

Ray traced

Path traced (130 spp)

Noon

Night



Rain

Annotations are “free”

Simulation Research – Problem Statement

FIAS Frankfurt Institute for Advanced Studies

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

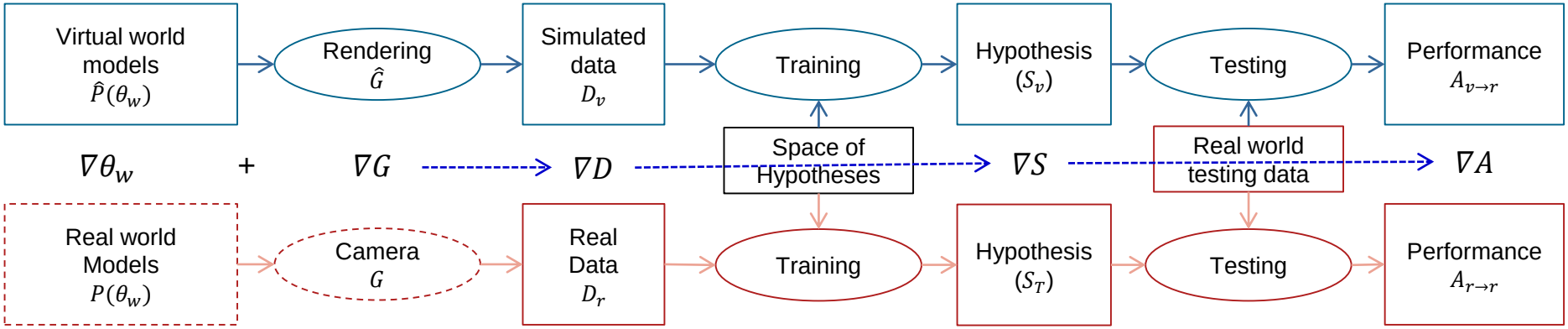
- **In 1990's, CV community had been skeptic to use CG to train.**
  - CG may use some mathematical simplifications and approximations that CV models based on. Hence, they might generate ideal or near ideal to CV models.
  - That was good question in the days of model-driven designs.

- **Recent video games may also use approximated models for realistic effects for the sake of interactive real time display.**
  - Now, in the days of data driven designs, how these approximations effects the CV ?

- **Deviations of Scene (parameters) distributions plus Physical accuracy of rendering processes contributes towards Domain-shift issue b/w virtual and real world data.**

- **Especially, the impact of modeling errors and computational rendering approximations, due to choices in the rendering and generation pipeline, on trained CV systems generalization performance is still not clear.**

Transfer and Domain shift



- No free lunch in the selection of $\hat{P}$ and $\hat{G}$ for data simulation processes.
- In principle, $\nabla\theta_w$ and ∇G impact the magnitudes of ∇D, ∇S, and ∇A.

- ***What is the impact of $\hat{G}$ on ΔA?***
  - Real time Photo-realism vs Expensive physics-realism?
- ***What is the impact of parameters of $\hat{P}(\theta_w)$ on ΔA?***
  - How far can we go with an arbitrary scene generative model?
  - Can unsupervised generative learning from target real data help?

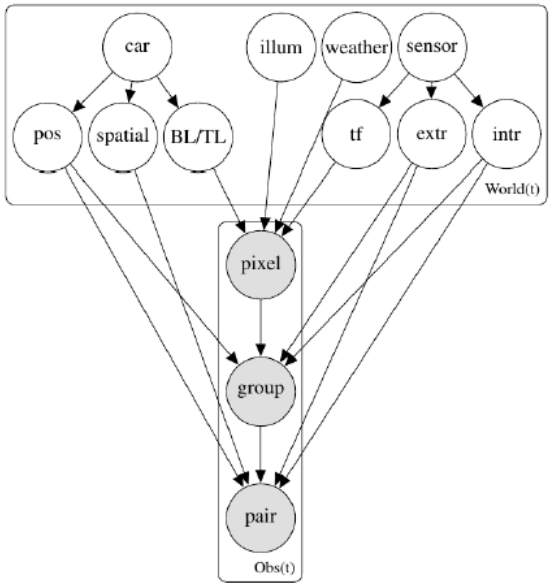
- **However, one can bypass these issues by simply adding some real samples to simulated data.**

Anomaly Detection – Case Study

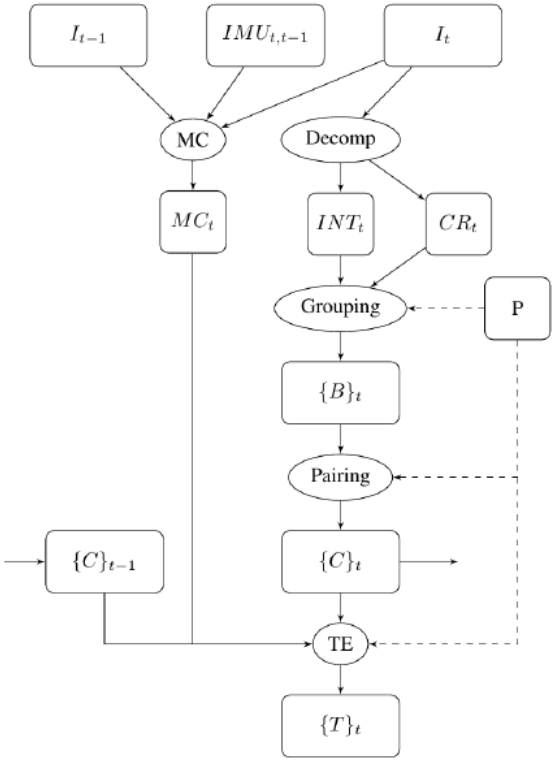
FIAS Frankfurt Institute for Advanced Studies

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Thesis / case study – Brakelight transition detection



car-pos	6d-pose of observed car
car-spatial	spatial layout of observed car's parts
car-BL/TL	Brakelight/Taillight characteristics (color, emission-strength)
illum	illumination
sensor-tf	transfer function
sensor-extr	extrinsic parameters
sensor-intr	intrinsic parameters
pixel	observed pixel values
group	spatial poolings of pixels
pairs	configurations of groups



Worldmodel:
- Latent variables
- Causal relations

Populate by
- Specifications
- Physics
- Simulations

Mapping to algorithms & pipeline:
- Identification of relations of latent contextual variables and observations -> submodalities
- C,T,P -> Modules + Parameters

Model-based Sys-engy approach:
- Loose coupling, single module evaluation
- Uncertainty propagation

Next Classes

- Introduction to Probabilistic Graphical Models
- Pattern Grammars and Inference

‘Thank you’ -- The End

Backup