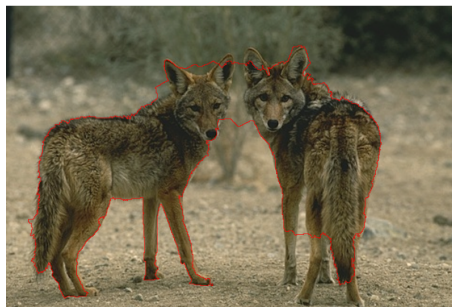


# BACHELOR THESIS

JAN LAMMEL



Untersuchung des Multicut Problems bei Verwendung einer Variation of  
Information Loss Funktion

September 2015

Jan Lammel: *Bachelor Thesis*, Untersuchung des Multicut Problems bei Verwendung einer Variation of Information Loss Funktion, © September 2015

## ABSTRACT

---

Short summary of the contents in English...a great guide by Kent Beck how to write good abstracts can be found here:

<https://plg.uwaterloo.ca/~migod/research/beck00PSLA.html>

## ZUSAMMENFASSUNG

---

Kurze Zusammenfassung des Inhaltes in deutscher Sprache...



# INHALTSVERZEICHNIS

---

1	EINLEITUNG	1
1.1	Motivation Variation of Information	1
2	THEORETISCHE GRUNDLAGEN	3
2.1	Graphen Theorie	3
2.2	Feature Space	3
2.3	Das Multicut Problem	3
2.4	Loss Funktionen	4
2.4.1	Partition Hamming	4
2.4.2	Variation of Information	4
2.5	Structured Learning	4
2.5.1	Subgradient Descent	5
2.5.2	Stochastic Gradient	5
3	EXPERIMENTELLES SETUP	7
3.1	Trainings- und Testdaten	7
3.2	Graphical Model Unterbau und Solver	7
3.3	Feature Space	7
4	EXPERIMENTE UND MESSDATEN	9
4.1	BSD500	9
I	APPENDIX	11
A	APPENDIX TEST	13
	LITERATURVERZEICHNIS	15



## EINLEITUNG

---

Das weite Forschungsfeld der Bildsegmentierung handelt von der Problemstellung, Bilder automatisiert in einzelne semantisch sinnvolle Segmente zu unterteilen. Anwendungsgebiete finden sich unter anderem in der Objekterkennung, Biologie, Medizin und allgemein Bildanalyse-Methoden, wobei die Segmentierung als Vorstufe zur weiteren Bearbeitung dient.

(Hier Bild einfügen: Bild ohne Segmentierung -> Bild mit gewünschter Segmentierung)

Dieser Prozess soll Lern-basiert sein, d.h. es werden dem Algorithmus Trainingsdaten mit Beispielbildern inklusive Soll-Segmentierung (Ground Truth) übergeben. Anhand dieser werden Parameter optimiert um möglichst allgemeingültig, den Trainingsdaten ähnliche Bilder ebenso in einzelne Segmente gliedern zu können.

In dieser Arbeit wird nun die neue Methode Variation of Information zur Quantifizierung der Qualität einer Segmentierung vorgestellt und mit einer bestehenden (Partition Hamming) verglichen. Da der Lern-Algorithmus auf diesem Kriterium aufbaut, ist dies elementar für die Güte der resultierenden Segmentierung.

### 1.1 MOTIVATION VARIATION OF INFORMATION

Bisher wird bei Partition Hamming jede einzelne Kante der Segmentierung überprüft, ob diese in der Ground Truth ebenso vorhanden ist oder nicht. Bei einer leicht verschobenen Segmentierung (siehe Bild) führt dies zu einem extrem hohen Loss, obwohl die Segmentierung nicht viel schlechter ist als die der Ground Truth. Dieser Fall kann beispielsweise eintreten wenn der Ground Truth Ersteller unsauber gearbeitet hat, was durchaus vorkommen kann wenn man davon etliche erstellt. Außerdem ist es oft Interpretationssache, wo nun genau eine Kante eines Objektes verläuft, wenn sich der zugehörige Gradient des Bildes über eine gewisse Fläche streckt.

Mit Variation of Information wird nun versucht, diese Nachteile zu beseitigen. Man betrachtet hierbei nicht mehr den Zustand jeder einzelnen Kante, ob diese nun an oder aus ist, sondern die einzelnen segmentierten Flächen. Die exakte mathematische Berechnung folgt in den theoretischen Grundlagen (2.4.2), anschaulich gesehen werden allerdings nur die Flächen bestraft, die ein unterschiedliches Label als die Ground Truth haben.

Bild veranschaulicht dies nun...



## 2.1 GRAPHEN THEORIE

Die Grundlage aller weiteren Betrachtungen ist ein Region Adjacency Graph (RAG). Um diesen zu erstellen, wird das Bild zunächst mithilfe des SLIC-Algorithmus (Zitat!) in Superpixel unterteilt, dessen Ränder möglichst an den Objektkonturen verlaufen. Das Ergebnis hiervon ist in (Abb verlinken) abgebildet.

Der Region Adjacency Graph  $G$  baut sich aus Nodes  $V$  und Edges  $E$  auf. In unserem Fall entsprechen die Nodes den Superpixeln. Die Edges bestehen nur zwischen denjenigen Nodes, bei denen die zugehörigen Superpixel direkt angrenzen und somit eine gemeinsame Kante besitzen.

(hier SLIC-partition Bild und RAG-Bild einfügen)

Bei der letztlichen Segmentierung geht es darum, ein konsistentes Labeling der Superpixel zu erreichen. Dies wird über die Aktivität der Edges erreicht, welche an- oder ausgeschaltet sein können. Für die Aktivität einer Edge  $y$  gilt somit:  $y \in \{0, 1\}$

Konsistent ist eine Segmentierung genau dann, wenn bei aktiven Edges die zugehörigen Superpixel verschiedene Labels haben und analog bei inaktiven Edges die Superpixel die gleichen Labels. Anschaulich gesehen ist dies der Fall, wenn alle aktiven Edges geschlossene Linien bilden.

## 2.2 FEATURE SPACE

Der Feature Space  $X \in \mathbb{R}^{|E| \times D}$  ordnet jeder Edge  $D$  Features zu, die möglichst in Korrelation zur Frage stehen, ob die betrachtete Edge nun aktiv oder inaktiv sein soll.

## 2.3 DAS MULTICUT PROBLEM

Anhand dieser gewichteten Edges eine konsistente Segmentierung zu erhalten wird als Multicut Problem (MP) bezeichnet. Es wird durch folgendes Minimierungsproblem beschrieben:

$$\begin{aligned} \arg \min_{y_e} \quad & \sum_{y \in E} w \beta_e \cdot y \\ \text{s.t.} \quad & y - \sum_{y_i \in P(y)} y_i \leq 0 \quad \forall y \in E \end{aligned} \quad (1)$$

Hierbei entspricht  $w \in \mathbb{R}^D$  den Weights und  $\beta_e \in \mathbb{R}^D$  den Funktionswerten der  $D$  extrahierten Informationen des Feature Spaces. Die Nebenbedingungen erzwingen die Konsistenz der Segmentierung.  $P(y)$  ist hierbei der kürzeste Pfad über inaktive Edges der beiden Superpixel, die benachbart zu  $y$  sind. In der Praxis wird das Minimierungsproblem zunächst ohne Constraints gelöst und anschließend solange für Edges hinzugefügt, die die Konsistenzbedingung verletzen, bis Konsistenz erreicht ist.

## 2.4 LOSS FUNKTIONEN

Mithilfe einer Loss Funktion  $\mathcal{L}(y, y^*)$  wird quantifiziert, wie gut eine Segmentierung  $y$  mit derjenigen der Ground Truth  $y^*$  übereinstimmt. In dieser Arbeit wird die neue Methode "Variation of Information" vorgestellt und mit der bestehenden "Partition Hamming" verglichen.

### 2.4.1 Partition Hamming

$$\mathcal{L}(y_i, y_i^*) = \begin{cases} \mathbb{I}[y_i \neq y_i^*] \cdot \alpha_{\text{over}} & \text{if } y_i^* = 0 \\ \mathbb{I}[y_i \neq y_i^*] \cdot \alpha_{\text{under}} & \text{if } y_i^* = 1 \end{cases} \quad \forall y_i \in E \quad (2)$$

$$\mathcal{L}(y, y^*) = \sum_{y_i \in E} \mathcal{L}(y_i, y_i^*) \quad (3)$$

Es werden direkt die Edges der Segmentierung  $y$  und der Ground Truth  $y^*$  verglichen und bei fehlender Übereinstimmung erhöht sich der Loss. Meist ist  $\alpha_{\text{under}} > \alpha_{\text{over}}$  um Übersegmentierung zu bevorzugen, da es tragischer ist Objekte nicht zu erfassen.

### 2.4.2 Variation of Information

$$\mathcal{L}(y, y^*) = H_y + H_{y^*} - 2 \cdot I(y, y^*) \quad (4)$$

$H_x$  ist hierbei die Entropie der Segmentierung  $x$ . Jede Segmentierung besitzt eine individuelle Entropie.

$I(x, x^*)$  bezeichnet die Transinformation, anschaulich gesehen entspricht diese der Schnittmenge der Ist- und Soll-Segmentierung. Es werden also die Labels der Superpixel untersucht und bei fehlender Übereinstimmung beim Vergleich mit der Ground Truth erhöht sich der Loss.

## 2.5 STRUCTURED LEARNING

Um später mithilfe des Multicut Algorithmus Bilder möglichst gut segmentieren zu können, muss der Parameter  $w$  bestimmt werden. "Möglichst gut" bedeutet hier in Bezug auf eine Loss Funktion, die als

Qualitätskriterium dient. Da ein niedriger Loss für eine gute Segmentierung steht ist also das folgende Minimierungsproblem zu lösen:

$$\hat{w} = \arg \min_w \mathcal{L}(y, y^*) \quad (5)$$

### 2.5.1 Subgradient Descent

Der Subgradient Descent Algorithmus basiert auf der Berechnung der Differenz der akkumulierten Feature der Segmentierung  $y$  und der Ground Truth  $y^*$ , welche gewichtet dem weight Vector  $w$  hinzuaddiert werden. Für nähere Details siehe [1].

Die Minimierung des Partition Hamming Losses in dieser Arbeit wird hiermit realisiert.

### 2.5.2 Stochastic Gradient

Der hier verwendete Stochastic Gradient ist eine Variante des in [1] näher erläuterten Algorithmus. Im folgenden wird die hier angewandte Methode zur Ermittlung der Gradientenrichtung (Alg. 1), sowie der Liniensuche (Alg. 2), also der Schrittweite pro Iterationsschritt beschrieben:

---

#### Algorithm 1 Get Gradient Descent Direction

---

```

1: procedure GETGRADIENTDESCENTDIRECTION( $n$ Perturbs,  $\sigma$ ,  $w$ )
2:    $\sigma$ : Noise standard deviation
3:    $w$ : Current Weight Vector
4:
5:    $\Delta x = 0$ 
6:   for  $n = 1 \dots n$ Perturbs do
7:     Generate Noise  $\in \mathcal{N}(0, \sigma^2)$  und add to  $w$ 
8:     Calculate Loss on current Training Sample
9:      $\Delta x = \Delta x + \text{Noise} * \text{Loss}$ 
10:   $\Delta x = -\Delta x / n$ Perturbs
11:  return  $\Delta x$ 

```

---

---

**Algorithm 2** Line Search and update Weights

---

```

1: procedure LINESEARCHANDTAKESTEP( $w, \eta, \Delta x$ )
2:    $\eta$ : Stepwidth
3:    $w$ : Current weight vector
4:    $\Delta x$ : Gradient Descent Direction
5:
6:   for  $n = \{0.1, 0.5, 1.0, 5.0, 10.0\}$  do
7:     Varied Weight Vector  $w_{\text{var}} = w + \Delta x \cdot n$ 
8:     Calculate mean Loss  $\mathcal{L}$  on entire Training Set
9:     from  $w_{\text{var}}$ 
10:    if  $\mathcal{L} < \mathcal{L}_{\text{best}}$  then
11:       $\mathcal{L}_{\text{best}} = \mathcal{L}$ 
12:      Save  $w_{\text{best}} = w$ 
13:      Break for-loop
14:    Memorize  $\mathcal{L}$  and associated varried Weight Vector
15:   $w = w_{\text{var}}$ , where regarding Loss is minimal
16:  return  $w$ 

```

---

## EXPERIMENTELLES SETUP

---

### 3.1 TRAININGS- UND TESTDATEN

Zum einen wurden Experimente an kleinen synthetisch erzeugten Bildern durchgeführt um die prinzipiellen Vorteile von Variation of Information zu demonstrieren.

Zum Anderen diente für die umfassenderen Experimente zur praktischen Anwendung das Berkeley Segmentation Dataset (BSD500) [2], welches aus natürlichen Bildern besteht. Es wurden hiervon die Test-Bilder genommen, da hierfür State of the Art Kantendetektoren als Feature zur Verfügung standen.

### 3.2 GRAPHICAL MODEL UNTERBAU UND SOLVER

Zur Generierung des Region Adjacency Graphs, des Random Forests und der Filter wurde VIGRA [5] verwendet. Inferno [6] zum Zusammenführen aller Daten, Lösen des Multicut Problems und Lernen der Parameter sowohl mit SubGradient bezüglich Partition Hamming, als auch mit Stochastic Gradient bezüglich Variation of Information.

### 3.3 FEATURE SPACE

Für die synthetischen Daten wurden folgende Feature gewählt:

- Gaussian Gradient Magnitude mit  $\sigma = 1$
- Hessian of Gaussian Eigenvalues mit  $\sigma = 1$
- Structure Tensor Eigenvalues

Beim BSD hat sich der Feature Space folgendermaßen zusammengesetzt:

- Gaussian Gradient Magnitude mit  $\sigma = \{1, 2, 5\}$
- Hessian of Gaussian Eigenvalues mit  $\sigma = 2$
- Laplacian of Gaussian
- Structure Tensor Eigenvalues
- Canny Filter
- $N^4$ -Fields Kantendetektor [3]

- Structured Forests Kantendetektor [4, Dollár et al.]
- Statistische Kenndaten in variablen Bereichen  $\bar{u}$  und  $\bar{v}$  um eine Edge an Superpixeln  $u$  und  $v$   
(seperat angewandt auf alle 3 Farbkanäle des eigentlichen Bildes, als auch auf den  $N^4$ -Fields- und Dollár-Kantendetektor)
  - $\text{Mean}(\bar{u} + \bar{v})$
  - $\text{Variance}(\bar{u} + \bar{v})$
  - $\frac{\max\{\text{Mean}(\bar{u}), \text{Mean}(\bar{v})\}}{\min\{\text{Mean}(\bar{u}), \text{Mean}(\bar{v})\}}$
  - $\frac{\max\{\text{Median}(\bar{u}), \text{Median}(\bar{v})\}}{\min\{\text{Median}(\bar{u}), \text{Median}(\bar{v})\}}$
  - $\text{Skewness}(\bar{u} + \bar{v})$
  - $\text{Kurtosis}(\bar{u} + \bar{v})$
- Konstantes Feature für jede Edge, zur Beseitigung des Bias im Feature Space

Zusätzlich wurde aus den Feature Spaces aller Trainingsdaten ein Random Forest (RF) aufgebaut und dieser zur Generierung eines weiteren Features verwendet.

## EXPERIMENTE UND MESSDATEN

## 4.1 BSD500

Fehler noch einfügen?!?! frage ob die std.-abweichung der losse sinn macht...

PH	VOI	PH Loss		VOI Loss	
		SubGrad	StochGrad	SubGrad	StochGrad
Feature $\in [-1, 1]$		<b>Test Set</b>			
mit RF Feature ohne Constraint	mit RF Feature Constraint auf RF	225.66	251.34	1.1587	1.2667
80 Iterationen $\eta = 0.1$	1 Iteration $\eta = 0.1$ $\sigma = 0.3$	<b>Training Set</b>			
		88.16	26.24	0.5190	0.1726
Feature $\in [-1, 1]$		<b>Test Set</b>			
mit RF Feature ohne Constraint	mit RF Feature ohne Constraint	225.66	280.51	1.1587	1.4009
80 Iterationen $\eta = 0.1$	1 Iteration $\eta = 0.1$ $\sigma = 0.3$	<b>Training Set</b>			
		88.16	24.95	0.5190	0.1460
Feature $\in [-1, 1]$		<b>Test Set</b>			
ohne RF Feature ohne Constraint	mit RF Feature Constraint auf RF	237.66	229.38	1.1653	1.1833
80 Iterationen $\eta = 0.1$	1 Iteration $\eta = 0.1$ $\sigma = 0.3$	<b>Training Set</b>			
		229.1	229.6	1.0678	1.0412
Feature $\in [-1, 1]$		<b>Test Set</b>			
ohne RF Feature ohne Constraint	mit RF Feature ohne Constraint	237.66	229.86	1.1653	1.1785
80 Iterationen $\eta = 0.1$	1 Iteration $\eta = 0.1$ $\sigma = 0.3$	<b>Training Set</b>			
		229.1	231.01	1.0678	1.0668

Tabelle 1: Messwerttabelle 1

		PH Loss		VOI Loss	
PH	VOI	SubGrad	StochGrad	SubGrad	StochGrad
Feature $\in [0, 1]$		<b>Test Set</b>			
mit RF Feature ohne Constraint	mit RF Feature Constraint auf RF	233.74	306.41	1.1663	1.4457
80 Iterationen	1 Iteration	<b>Training Set</b>			
$\eta = 0.1$	$\eta = 0.1$	82.28	41.58	0.4919	0.2558
	$\sigma = 0.3$				
Feature $\in [0, 1]$		<b>Test Set</b>			
mit RF Feature ohne Constraint	mit RF Feature ohne Constraint	233.74	248.85	1.1663	1.2847
80 Iterationen	1 Iteration	<b>Training Set</b>			
$\eta = 0.1$	$\eta = 0.1$	82.28	42.83	0.4919	0.2585
	$\sigma = 0.3$				
Feature $\in [0, 1]$		<b>Test Set</b>			
ohne RF Feature ohne Constraint	mit RF Feature Constraint auf RF	237.18	246.89	1.1966	1.2711
80 Iterationen	1 Iteration	<b>Training Set</b>			
$\eta = 0.1$	$\eta = 0.1$	228.94	187.93	1.0828	0.9523
	$\sigma = 0.3$				
Feature $\in [0, 1]$		<b>Test Set</b>			
ohne RF Feature ohne Constraint	mit RF Feature ohne Constraint	237.18	237.49	1.1966	1.2207
80 Iterationen	1 Iteration	<b>Training Set</b>			
$\eta = 0.1$	$\eta = 0.1$	228.94	82.67	1.0828	0.4775
	$\sigma = 0.3$				

Tabelle 2: Messwerttabelle 2

Einzelne Bilder rauspicken mit spez. Loss-Angabe um zu zeigen, dass Untersegmentierung nach VOI geringen loss ergibt...



Teil I

APPENDIX







[?]



## LITERATURVERZEICHNIS

---

- [1] Sebastian Nowozin, Christoph H. Lampert, *Structured Learning and Prediction in Computer Vision*, 2011.
- [2] D. Martin, C. Fowlkes, D. Tal and J. Malik, *A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics*, Proc. 8th Int'l Conf. Computer Vision, July 2001
- [3] aroslav Ganin and Victor Lempitsky, *N<sup>4</sup>-Fields: Neural Network Nearest Neighbor Fields for Image Transforms*, Skolkovo Institute of Science and Technology, 2014
- [4] Piotr Dollár and C. Lawrence Zitnick, *Structured Forests for Fast Edge Detection*, ICCV 2013
- [5] Ullrich Köthe, *Vision with Generic Algorithms*, Image Processing and Analysis Library, Version 1.10.0
- [6] Thorsten Beier





## DECLARATION

---

Put your declaration here.

*Saarbrücken, September 2015*

---

Jan Lammel