

# Adaptive Reconstruction of Discrete-Valued Objects from few Projections

Thomas Schüle <sup>a,b,1</sup>, Stefan Weber <sup>a,2</sup>, Christoph Schnörr <sup>a,3</sup>

<sup>a</sup> *Dept. M&CS, University of Mannheim, D-68131 Mannheim, Germany*

<sup>b</sup> *Siemens Medical Solutions, D-91301 Forchheim, Germany*

---

## Abstract

Recently, we proposed an algorithm for binary tomography based on DC ( difference of convex functions ) programming [13,15]. In this paper, we extend the binary reconstruction problem to multi-valued objects. We describe how such objects can be reconstructed just by combining binary decisions. The proposed algorithm remains practicable for multi-valued reconstructions, and even with a large number of discrete values. Furthermore, we show how approximately known absorption levels can be adaptively estimated within the reconstruction process.

*Keywords:* Discrete Tomography, Combinatorial Optimization, Concave Minimization, D.C. Programming, EM-Algorithm

---

## 1 Introduction

Material and body tissue can be modeled in many cases with a limited set of classes, so binary or discrete valued objects arise naturally in industrial

---

<sup>1</sup> Email: [schuele@uni-mannheim.de](mailto:schuele@uni-mannheim.de)

<sup>2</sup> Email: [wstefan@uni-mannheim.de](mailto:wstefan@uni-mannheim.de)

<sup>3</sup> Email: [schnoerr@uni-mannheim.de](mailto:schnoerr@uni-mannheim.de)

and medical applications. To get information about the distribution inside such objects, transmission tomography has gained a wide-spread use as non-destructive technique. Here the absorptions of X-rays passing through an object are measured for a set of projection angles. Unfortunately standard non-discrete reconstruction techniques such as filtered back-projection need a very large amount of projections over an angle covering more than  $180^\circ$  in order to achieve usable results. In contrast a typical application of discrete tomography is the reconstruction of discrete valued objects from only a few projections with a limited scanning angle geometry. These scenarios arise because of technical limitations or because the object is very sensitive against radiation. The main topic in discrete tomography is to use the prior knowledge of discrete valued objects to achieve better reconstruction results on such scenarios.

## 2 Related Work

Based on the work in [15] we extend our approach to reconstructions with more than two values. Our mathematical programming method provides an alternative to sequential update techniques and other iterative reconstruction methods, see e.g. [5,9]. We incorporate prior information based on Markov-Random-Field models without using stochastic sampling methods [2,11]. The adaptive estimation of absorption coefficients is motivated as hidden parameter model in the context of expectation maximization (EM), see e.g. [16] and references therein for EM in discrete tomography.

## 3 Binary Objects

### 3.1 Reconstruction

Using a linear transformation for the projection process of transmission tomography, the reconstruction problem becomes an inverse problem for

$$(1) \quad Ax = b$$

with unknown  $x \in \{0, 1\}^n$  for the binary reconstruction problem. Elements of the matrix  $A$  ( with  $a_{ij} \geq 0$  ) represent the portion of a X-ray through the discretized subspace of the object domain, e.g. a pixel element in 2D or a volume element in 3D.

We model the binary reconstruction problem as an optimization problem

$$(2) \quad \min_{x \in \{0,1\}^n} f(x) := \frac{1}{2} (Ax - b)^2.$$

This integer program is a combinatorial problem and algorithms especially for large scale systems are computationally demanding. Therefore we use a variational formulation with relaxed  $x \in [0, 1]^n$ . Due to the ill-posedness of the inverse problem an improvement of the reconstruction quality can be achieved with an regularization term, which models prior information about the properties of an object. We use a smoothness prior to penalize non-homogeneous and isolated areas:

$$(3) \quad \min_{x \in [0,1]^n} E(x) := f(x) + \alpha \langle x, Lx \rangle$$

$$(4) \quad = x^T Qx + \langle q, x \rangle + c,$$

with  $L$  describing the difference of neighboring elements and with the regularization parameter  $\alpha$ . This yields

$$(5) \quad \langle x, Lx \rangle := \sum_{\langle i,j \rangle} (x_i - x_j)^2,$$

where  $\langle i, j \rangle$  ranges over the edges of the underlying grid graph.

### 3.2 Convex-Concave Regularization

The drawback of the variational formulation in (3) is the relaxation of solutions to non-binary values. Hence a thresholding scheme is needed to obtain binary or discrete-valued results. To avoid this post-processing step we enrich the functional with a concave part to drive the solution to binary values:

$$(6) \quad \min_{x \in [0,1]^n} F(x; \lambda) := E(x) + \lambda \langle x, e - x \rangle.$$

By starting with  $\lambda = 0$  and increasing  $\lambda$  slowly we end up in binary values. Note that we have now a functional  $\min_{x \in [0,1]^n} F(x; \lambda)$  with both convex and concave parts. To solve this constrained optimization problem, we use the DC ( difference of convex functions ) programming approach.

## 4 DC Programming

In this section we give a sketch of DC ( difference of convex functions ) programming and how the optimization of  $F(x; \lambda)$  is related to DC. For more details on DC, we refer to [13,15].

By decomposing  $F(x; \lambda)$  into a convex and a concave part, we model our problem as d.c. program, which is defined as the following minimization problem [13]

$$(7) \quad \alpha = \inf \{ f(x) := g(x) - h(x), x \in X \}$$

with proper lower semicontinuous convex functions  $g(x)$  and  $h(x)$  on  $X$ . To solve (7), the simplified DC-algorithm uses a primal-dual iteration scheme [13]:

**Algorithm 1 DC-Algorithm (DCA):**

- Choose  $x^0 \in \text{dom } g$  arbitrary.
- For  $k = 0, 1, \dots$  compute:
  - (8)  $y^k \in \partial h(x^k)$ ,
  - (9)  $x^{k+1} \in \partial g^*(y^k)$ .

Here,  $g^*$  denotes the Fenchel-conjugate of (7) and  $\partial$  denotes the subgradient, see [13].

## 5 Binary Reconstruction Algorithm

The decomposition of  $F(x; \lambda)$  into two convex functions is not unique. In order to investigate how reconstruction performance and efficiency depends on the decomposition, we choose two different decompositions. The crucial part is step (9). because there we have to solve a constrained optimization problem which is not trivial for large scale problems. To keep this step as simple as possible, we choose the following first decomposition

**Decomposition I:**

$$(10) \quad g(x) = \frac{1}{2} \langle x, \tau I x \rangle + \delta_C(x), \quad C = [0, 1]^n,$$

$$(11) \quad h(x; \lambda) = \frac{1}{2} \langle x, (\tau I - Q)x \rangle - \langle q, x \rangle - \frac{1}{2} \lambda \langle x, (e - x) \rangle$$

$$(12) \quad = \frac{1}{2} \langle x, [(\tau + \lambda)I - Q]x \rangle - \langle q + \frac{1}{2} \lambda e, x \rangle,$$

where  $\tau$  is chosen sufficiently large to guarantee convexity of  $h$ .

Note that both  $g$  and  $h$  are convex. Since  $h$  is smooth,  $\partial h(x) = \{\nabla h(x)\}$  and step (8) amounts to evaluate the gradient:

$$(13) \quad y^k = \nabla h(x^k; \lambda)$$

$$(14) \quad = [(\tau + \lambda)I - Q]x^k - (q + \frac{1}{2} \lambda e).$$

Function  $g$ , on the other hand, is non-smooth due to the constraint  $x \in C$ , and we have to solve problem:

$$(15) \quad x^{k+1} \in \partial g^*(y^k)$$

$$(16) \quad = \operatorname{argmin}_x \{g(x) - \langle y^k, x \rangle\}.$$

Starting with a convex functional for  $\lambda = 0$ , we increase  $\lambda$  to enforce binary solutions. This yields the following algorithm [15]

**Algorithm 2 Binary Reconstruction Algorithm:**

- Choose  $\lambda^0 = 0$ ,  $x^0 = \frac{1}{2}e$ ,  $\Delta_\lambda$
- Repeat ( outer DC-loop )
  - Repeat ( inner DC-loop )
    - $y^k = \nabla h(x^k; \lambda)$
    - $x^{k+1} = \operatorname{argmin}_x \{g(x) - \langle y^k, x \rangle\}$
    - until  $|x^{k+1} - x^k| \leq \epsilon$
    - $\lambda^{l+1} = \lambda^l + \Delta_\lambda$
- until  $x_i^{k+1} \notin [\epsilon', 1 - \epsilon'] \forall i$

Note that (16) can be calculated analytically for decomposition I, see [15]. To increase convergence speed, other decompositions are plausible. The matrix  $Q$  is positive semi-definite which suggests to use the following second decomposition

**Decomposition II:**

$$(17) \quad g(x) = \frac{1}{2} \langle x, Qx \rangle + \langle q, x \rangle + \delta_C(x), \quad C = [0, 1]^n,$$

$$(18) \quad h(x; \lambda) = -\frac{1}{2} \lambda \langle x, (e - x) \rangle.$$

The optimization problem in step (16) is now a constrained quadratic program, which can be solved with standard quadratic program solvers. Hence the binary reconstruction algorithm solves a sequence of quadratic programs which converge to a binary solution. As we gain convergence speed in the outer loop, each inner loop step needs to calculate (16), which may be computationally very demanding for large scale problems. For this kind of problems, decomposition I seems to be more convenient, as we can use sparse matrix multiplications to calculate a solution.

To show the convergence properties of both decompositions, we simulated 3 projections over 90 degree of a  $20 \times 20$  image ( see figure 1 ). Both decompositions, I and II, were able to reconstruct the original image in this case. Decomposition II was solved with CPLEX 7.5, a commercial quadratic program solver. In figure 1 we plotted the inner iteration loop for  $\lambda = 0$  against  $E(x)$ . Note that decomposition II shows faster convergence in this example, but keep in mind that each iteration step is computationally more costly. Figure 1 shows also the outer loop for both decompositions where the

quadratic programming approach ( decomposition II ) achieves better local minima during the iteration loop.

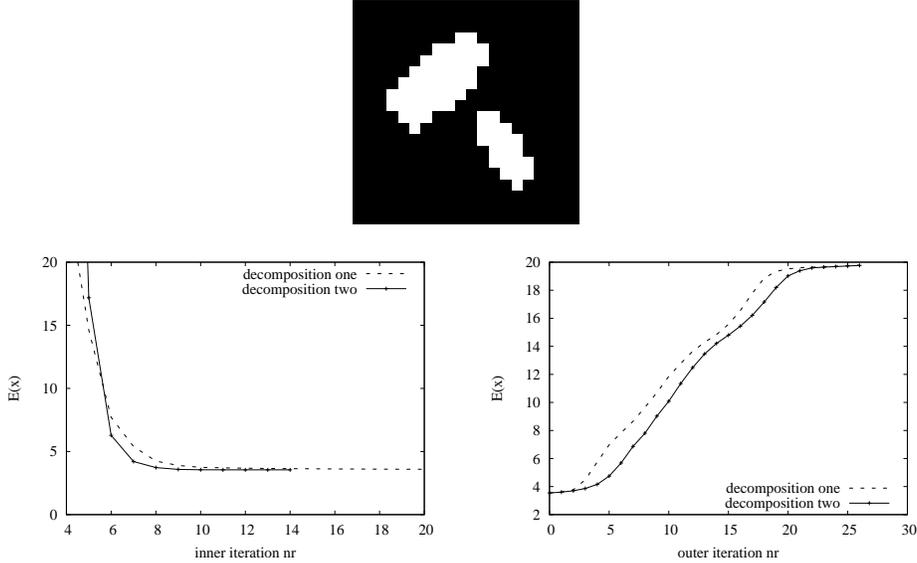


Fig. 1. Top: Ground truth of reconstruction problem. Bottom left: Convergence of inner DC loop for  $\lambda = 0$  for decomposition I and II. Bottom right: Developing of  $E(x)$  for outer DC loop for decomposition one and two.

## 6 From Binary to Discrete-Valued Objects

The basic idea of our discrete-valued reconstruction algorithm is to represent a non-binary object consisting of  $n + 1$  different absorption coefficients with binary variables. Therefore we define for each variable  $i$ , which may correspond to a pixel or voxel, the true or physical absorption  $\tilde{x}_i$  associated to this variable. Then the absorption  $\tilde{x}_i$  of  $i$  can be represented as linear combination with coefficients from a set  $M := \{\mu_0, \dots, \mu_n\}$  of all possible absorption coefficients. This yields

$$(19) \quad \tilde{x}_i = x_i \cdot \mu_{i,t} + (1 - x_i) \cdot \mu_{i,t-1},$$

with  $\mu_{i,t} \in M$  and  $\mu_{t-1} \leq \mu_t$ . Index  $t \in \{0, \dots, n\}$  in (19) picks the corresponding absorption coefficient  $\mu_{i,t} \in M$  for variable  $i$ .

Note that for  $x_i \in \{0, 1\}$  we get  $\tilde{x}_i \in \{\mu_0, \dots, \mu_n\}$ , i.e all pixels or voxels have discrete values. Based on this representation, we define the following

optimization problem

$$(20) \quad \min_{x \in [0,1]^n} F'(x; \lambda, \mu_t) := E(\tilde{x}) + \lambda \langle x, e - x \rangle,$$

with  $\tilde{x}$  as defined in (19) and by increasing  $\lambda$  we obtain discrete-valued solutions.

### 6.1 Convex Solution

For  $\lambda = 0$  the problem (20) is convex and we do not have to calculate a specific binary representation for  $\tilde{x}_i$ . We solve

$$(21) \quad \min_{x \in [0,1]^n} F'(x; 0) = E(\tilde{x})$$

by setting

$$(22) \quad \tilde{x}_i = x_i \cdot \mu_n + (1 - x_i) \cdot \mu_0$$

where  $\mu_0$  and  $\mu_n$  represent the lowest and highest absorption level respectively. Note that we can use the binary DC decomposition to calculate this solution.

### 6.2 Bin Update Step

In this step we determine  $\mu_{i,t-1}$  and  $\mu_{i,t}$  for each variable  $i$ . Therefore we identify in which half-open interval  $\text{bin}_t := [\mu_{t-1}, \mu_t)$  the current iterate  $\tilde{x}_i^{k-1}$  is located and represent it as linear combination of  $\mu_{t-1}$  and  $\mu_t$ , which are the corresponding bin borders:

**Bin Update Step:**

If  $\tilde{x}_i^{k-1} \in \text{bin}_t$

set  $\mu_{i,t-1} = \mu_{t-1}$ ,  $\mu_{i,t} = \mu_t$  and calculate  $x_i$  such that:

$$\tilde{x}_i^{k-1} = x_i \cdot \mu_{i,t} + (1 - x_i) \cdot \mu_{i,t-1}.$$

If  $\tilde{x}_i^{k-1} = \mu_n$

use  $\text{bin}_n$  for representation.

Note that we have just chosen another representation of the current iterate  $\tilde{x}_i^{k-1}$  in terms of the two absorption levels  $\mu_{i,t}$  and  $\mu_{i,t-1}$  associated with each variable  $x_i$ .  $\tilde{x}_i^{k-1}$  itself remains unchanged. Furthermore, we have not introduced any further variables to reconstruct more than two values at each pixel  $i$ . We still use the binary model independent from the number of discrete levels.

### 6.3 Bin Border Step

The bin definition above is not symmetric, i.e. values of  $x_i \in \text{bin}_t$  may converge into  $\text{bin}_{t+1}$  during the iteration, but not into  $\text{bin}_{t-1}$ . Consequently we iteratively change the bin borders as follows

**Bin Border Step:**

for iteration k odd define  $\text{bin}_t$  as:

$$\text{bin}_t := [\mu_{t-1}, \mu_t)$$

for iteration k even define  $\text{bin}_t$  as:

$$\text{bin}_t := (\mu_{t-1}, \mu_t]$$

### 6.4 Discrete DC-Algorithm

We insert the bin update and bin border step in the DC loop to reconstruct non-binary objects within the DC framework:

**Algorithm 3 Discrete DC-Algorithm:**

- *Step0 - calculate a convex solution*
- *Repeat*
  - *bin border step*
  - *bin update step*
  - $\arg \min_{x \in [0,1]^n} F'(x; \lambda)$
  - *increase  $\lambda$*
- *until  $x_i^{k+1} \notin [\epsilon', 1 - \epsilon'] \forall i$*

A reconstruction example for the discrete-valued algorithm is given in figure 2. Although the convex solution is far from the original image, the reconstruction shows how convex-concave regularization increases the reconstruction quality also for the multi-valued case.

## 7 Adaption of Absorption Coefficients ( Binary Case )

For real world problems absorption coefficients may be known only approximately due to noise or inhomogeneous distributions inside an object. Using the expectation-maximization (EM) algorithm, we describe how these absorption coefficients can also be estimated as hidden variables and adapted to increase the reconstruction quality. Note that we focus in this paper on the two-valued case, but an extension to discrete-valued reconstructions is possible, in principle.

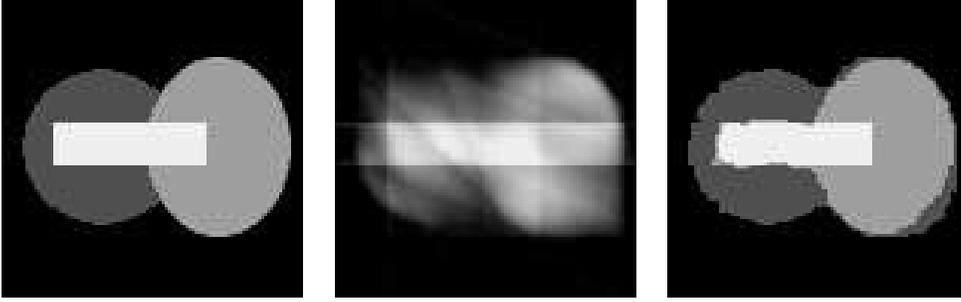


Fig. 2. Reconstruction result of a  $100 \times 100$  image consisting of 4 different classes ( left:ground truth ) with 5 projections over  $90^\circ$ . The reconstruction of the convex quadratic functional ( $\lambda = 0$ ) is shown in the middle. The right image displays the reconstruction result of our new discrete-valued reconstruction algorithm.

### 7.1 Two-Valued Reconstruction Problem

Lets express the optimization problem (6) in terms of the EM algorithm. Then our objective is to compute the maximum a posteriori reconstruction given by

$$(23) \quad p(x|b) \propto p(b|x)p(x),$$

where

$$(24) \quad p(b|x) \propto \exp\left(-\frac{1}{2}(Ax - b)^T(Ax - b)\right),$$

$$(25) \quad p(x) \propto \exp\left(-\alpha\langle x, Lx \rangle - \frac{1}{2}\lambda\langle x, e - x \rangle\right).$$

Now consider the following general two-valued reconstruction problem

$$(26) \quad Ax\mu_1 + A(e - x)\mu_0 = b,$$

with  $x \in \{0, 1\}^n$ . For this problem the projection values  $b$  are given and we want to find a reconstruction of  $x$  with *scalar parameters*  $\mu_0$  and  $\mu_1$ . These parameters define the unknown absorption coefficients of the two classes to reconstruct and they are the missing ( or hidden ) data in terms of EM.

### 7.2 EM-Algorithm

The expectation-maximization (EM) algorithm converges to a local maximum of the a posteriori function  $p(x|b)$  by including and estimating the missing ( or hidden ) absorption parameters  $\mu = (\mu_0, \mu_1)^T$  via marginalization [10]

$$(27) \quad p(b|x) = \int_{\mu} p(\mu, b|x)d\mu,$$

$$(28) \quad = \int_{\mu_0} \int_{\mu_1} p(\mu_0, \mu_1, b|x) d\mu_1 d\mu_0.$$

Then the expectation value of the complete data log-likelihood  $\log p(b, \mu|x)$  is given by

**E-step:**

$$(29) \quad Q(x, x^{(k-1)}) = \log p(x) + E_{\mu}[\log p(b, \mu|x)|b, x^{(k-1)}]$$

$$(30) \quad = \log p(x) + \int_{\mu} \log p(b, \mu|x) p(\mu|b, x^{(k-1)}) d\mu,$$

and for the maximization step we calculate

**M-step:**

$$(31) \quad x^{(k)} = \arg \max Q(x, x^{(k-1)}).$$

### 7.3 E-Step

In the appendix we show how  $Q$  can be defined as

$$(32) \quad Q := F^*(x; \hat{\mu}, \hat{\sigma}, \hat{\rho}, \lambda) := \alpha(\langle x, Lx \rangle) + \lambda \langle x, e - x \rangle - (c_0(\hat{\mu}_1^2 + \hat{\sigma}_1^2) + c_1 \hat{\mu}_1$$

$$(33) \quad + c_2(\hat{\mu}_0^2 + \hat{\sigma}_0^2) + c_3(\hat{\mu}_0)$$

$$(34) \quad + c_4(\hat{\mu}_0 \hat{\mu}_1 + \hat{\rho} \hat{\sigma}_0 \hat{\sigma}_1) + c_5),$$

where  $c_0, \dots, c_5$  are linear and quadratic terms of  $x$  and  $b$  ( see the appendix for details ) and  $\hat{\mu}_1$  and  $\hat{\mu}_0$  are the estimated  $\mu$  parameters.  $\hat{\mu}_0, \hat{\mu}_1, \hat{\sigma}_0, \hat{\sigma}_1$  and  $\hat{\rho}$  are calculated in each E-step based on  $x^{(k-1)}$  as described in the appendix.

### 7.4 M-Step

The M-step is given by

$$(35) \quad \min_{x \in [0,1]} F^*(x; \hat{\mu}, \hat{\sigma}, \hat{\rho}, \lambda),$$

which can be calculated with the binary DC decomposition. Now we embed the EM step inside the DC loop which yields the following algorithm

**Algorithm 4 Adaptive DC-Algorithm:**

- Choose  $\hat{\mu}^{(0)}, \hat{\sigma}^{(0)}$  and set  $\lambda^{(0)} = 0$
- Repeat (DC-loop)
  - Repeat (EM-loop)
    - E-step* : calculate  $\hat{\mu}^k, \hat{\sigma}^k$  and  $\hat{\rho}^k$
    - M-step* :  $x^{(k+1)} = \operatorname{argmin}_{x \in [0,1]} F^*(x, \hat{\mu}^k, \hat{\sigma}^k, \hat{\rho}, \lambda^{(l)})$
  - until  $|x^{(k+1)} - x^{(k)}| \leq \epsilon$  (EM-loop)
  - $\lambda^{(l+1)} = \lambda^{(l)} + \Delta_\lambda$
- until  $x_i^{k+1} \notin [\epsilon', 1 - \epsilon'] \forall i$  ( DC-loop )

In figure 3 we give a numerical example which shows the result without and with adapting the absorption coefficients. Without adaption, the background is modeled here as object in order to fulfill the projection constraints. In contrast, the adaptive DC algorithm was able to adjust  $\mu$  which yields a better reconstruction.

## 8 Numerical experiments

### 8.1 Discrete-Valued Reconstruction

We used a  $100 \times 100$  Shepp-Logan phantom ( figure 4 ) to test the discrete-valued algorithm which was implemented with decomposition one. The ground truth image has 6 different pixel values, namely

$$(36) \quad \mu = \{\mu_0, \dots, \mu_5\} = \{0.0, 0.1, 0.2, 0.3, 0.4, 1.0\}.$$

Since these absorption levels are not equally spaced, a direct usage of the smoothness prior  $\langle \tilde{x}, L\tilde{x} \rangle$  tends to over-smooth elements with  $\mu_5$  whereas  $\mu_0, \dots, \mu_4$  are underregularized. As a result we use the following mapping

$$(37) \quad \mu = \{\mu_0, \dots, \mu_N\} \rightarrow \bar{\mu} = \left\{0, \frac{1}{N}, \frac{2}{N}, \dots, 1\right\}$$

to model smoothness according to classification. This yields the prior  $\langle \bar{x}, L\bar{x} \rangle$  with

$$(38) \quad \bar{x}_i = x_i \cdot \bar{\mu}_{i,t} + (1 - x_i) \bar{\mu}_{i,t-1}.$$

We simulated 8,16, 24, 32 and 64 projections over 180 degree using parallel projection. The calculation time was about 600s with research code on a standard 3 GHz Pentium 4 PC. The numerical results in figure 4 demonstrate that our new approach is also applicable to quite complex objects.

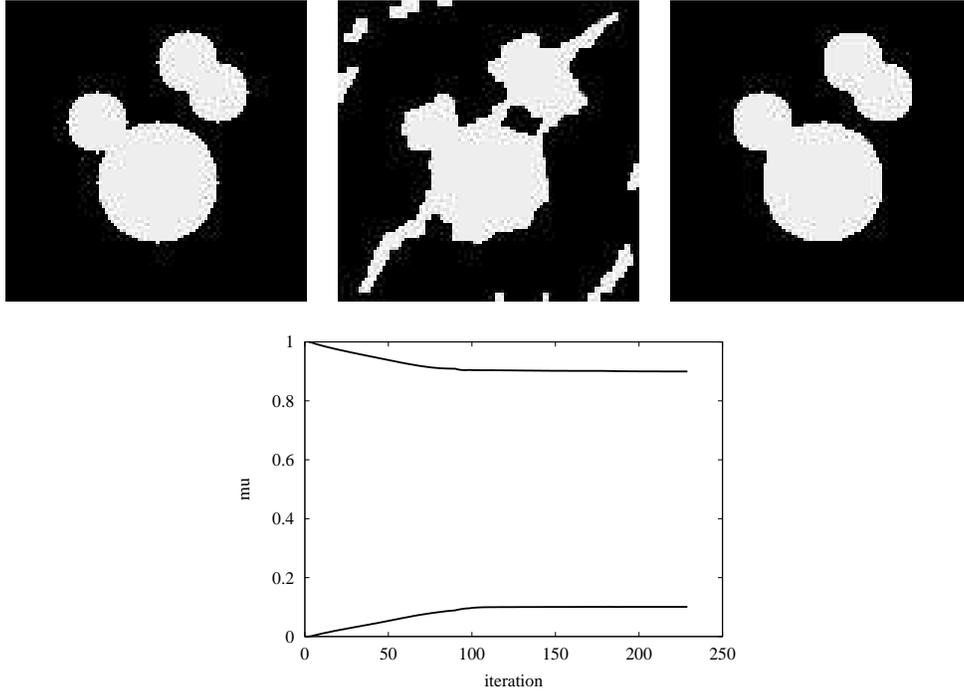


Fig. 3. Top left: Ground truth with two absorption levels of 0.1 ( black ) and 0.9 ( white ). We used 5 projections over  $90^\circ$  for reconstruction. Top middle: Binary DC reconstruction with fixed absorption estimations of 0.0 and 1.0. Top right: Result by adapting the absorption levels. Bottom: Convergence of absorption levels from the initial estimate to ground truth values of 0.1 and 0.9 for this experiment. The upper line plots the object value ( white ), whereas the lower line plots the background value ( black ).

### 8.2 Estimation of Absorption $\mu$

To show the robustness of our adaptive algorithm we used different kinds of noise to model inhomogeneous absorption. Figure 5, 6 and 7 shows that the absorption estimation is also usable for low signal to noise ratios with uniform, Gaussian and salt and pepper noise.

## 9 Conclusion and Further Work

In this work we have shown that the extension of our binary reconstruction algorithm to reconstructions of multi-valued objects via a binning step is possible. Convex-concave regularization improves the reconstruction quality also in case of discrete-valued objects. Furthermore we used the EM-algorithm to

motivate the adaption of the absorption coefficients as hidden data estimation and embedded this idea in our DC approach.

The smoothness prior used in this paper tends to over-smooth thin structures, like the outer white ellipse in the Shepp-Logan case. Hence more sophisticated priors should be used to further improve the reconstruction quality. From an application point of view a wide range of alternative decompositions are investigable to increase convergence speed without introducing a high computational burden.

## References

- [1] Censor, Y., *Binary steering in discrete tomography reconstruction with sequential and simultaneous iterative algorithms*, Lin. Algebra and its Appl. **339** (2001), 111–124.
- [2] Chan, M. T., G. T. Herman, and E. Levitan, *Bayesian image reconstruction using image-modeling gibbs priors*, Int. J. Imag. Syst. Technol. **9** (1998), 85–98.
- [3] Demoment, G., *Image reconstruction and restoration: Overview of common estimation structures and problems*, IEEE Trans. on Acoustics, Speech, and Signal Proc. **37(12)** (1989), 2024–2036.
- [4] Fishburn, P., P. Schwander, L. Shepp, and R. Vanderbei, *The discrete radon transform and its approximate inversion via linear programming*, Discr. Appl. Math. **75** (1997), 39–61.
- [5] Frese, T., C. A. Bouman, and K. Sauer, *Multiscale bayesian methods for discrete tomography.*, in G. T. Herman and A. Kuba, editors, “Discrete Tomography: Foundations, Algorithms, and Applications”, pages 237–264, Birkhäuser, Boston, 1999.
- [6] Gardner, R.J., and P. Gritzmann, *Discrete tomography: Determination of finite sets by x-rays*, Trans. Amer. Math. Soc. **349(6)** (1997), 2271–2295.
- [7] Gritzmann, P., D. Prangenberg, S. de Vries, and M. Wiegelmann, *Success and failure of certain reconstruction and uniqueness algorithms in discrete tomography.*, Int. J. Imag. Syst. Technol. **9** (1998), 101–109.
- [8] Gritzmann, P., S. de Vries, and M. Wiegelmann, *Approximating binary images from discrete x-rays*, SIAM J. Optimization **11(2)** (2000), 522–546.
- [9] Herman, G.T., and A. Kuba, “Discrete Tomography: Foundations, Algorithms, and Applications”, Birkhäuser, Boston, 1999.

- [10] MacLachlan, G., and T Krishnan, “The EM Algorithm and Extensions”, Wiley, New York, 1997.
- [11] Matej, S., G. T. Herman, and A. Vardi, *Binary tomography on the hexagonal grid using gibbs priors*, Int. J. Imag. Syst. Technol. **9** (1998), 126–131.
- [12] Natterer, F., and F. Wübbeling, *Mathematical Methods in Image Reconstruction*, SIAM, 2001.
- [13] Pham Dinh, T., and L. T. Hoai An, *A d.c. optimization algorithm for solving the trust-region subproblem*, SIAM J. Optim. **8(2)** (1998), 476–505.
- [14] Rockafellar, R. T., “Convex analysis”, Princeton Univ. Press, Princeton, NJ, 2nd. edition, 1972.
- [15] Schüle, T., C. Schnörr, S. Weber, and J. Hornegger, *Discrete Tomography by Convex-Concave Regularization and D.C. Programming*, Technical Report 15/2003, Computer Science Series, University of Mannheim, Dec. 2003. To appear in Discrete Applied Mathematics, Elsevier.
- [16] Vardi, Y., and C. H. Zhang, *Reconstruction of Binary Images from Limited Projections via the EM Algorithm*, in G. T. Herman and A. Kuba, Eds., “Discrete Tomography: Foundations, Algorithms and Applications”, pp.297-316, Birkhauser, Boston, 1999 .
- [17] Weber, S., T. Schüle, C. Schnörr, and J. Hornegger, *A Linear Programming Approach to Limited Angle 3D Reconstruction from DSA Projections*, Special Issue of Methods of Information in Medicine, 4, Schattauer Verlag, pages 320-326, 2004.
- [18] Weber, S., C. Schnörr, and J. Hornegger, *A linear programming relaxation for binary tomography with smoothness priors*, in Proc. Int. Workshop on Combinatorial Image Analysis (IWCLIA’03), Palermo, Italy, May 14-16 2003.

## A Appendix

### A.1 E-Step

Assuming normal distributions for the projection values we can define the following distribution

$$(A.1) \quad p(b|\mu, x) = \frac{1}{\sqrt{(2\pi)^n}} e^{-0.5(b-(Ax\mu_1+A(e-x)\mu_0))^T(b-(Ax\mu_1+A(e-x)\mu_0))}.$$

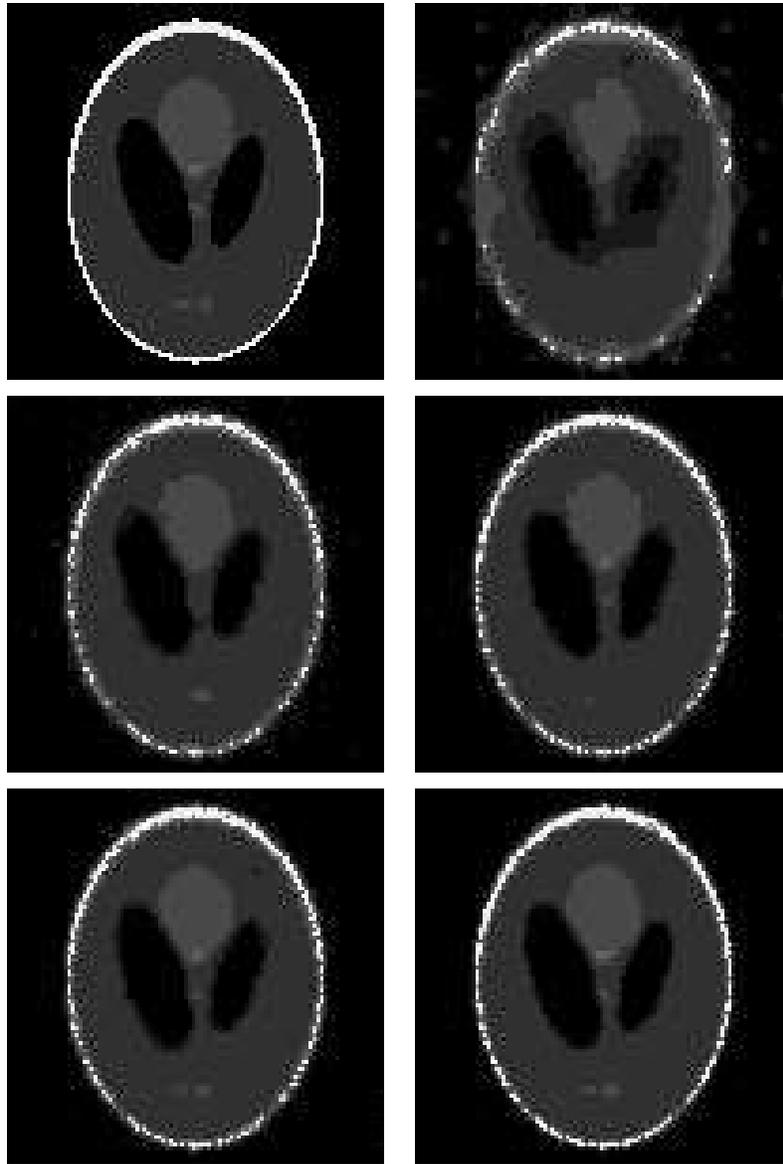


Fig. 4. Top left: Ground truth  $100 \times 100$  Shepp-Logan phantom. Top right: Reconstruction result using 8 projections over 180 degree. Middle left: Reconstruction with 16 projections. Middle right: Reconstruction with 24 projections. Bottom left: Reconstruction with 32 projections. Bottom right: Reconstruction with 64 projections.

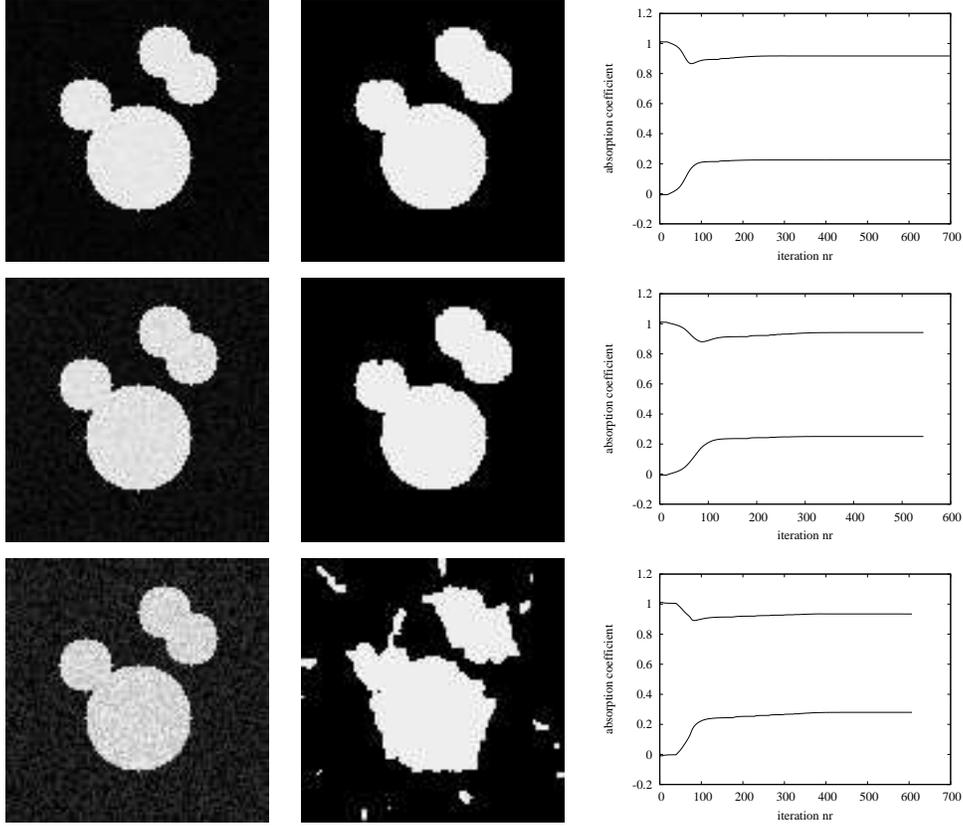


Fig. 5. Uniform noise: Reconstruction result with uniform additive noise for a  $100 \times 100$  image using 5 projections over 180 degree. Left column: Ground truth with noise and  $\mu_0 = 0.2$  and  $\mu_1 = 0.9$  Center column: Reconstruction result with adaptive DC-algorithm Right column: Developing of absorption coefficients. Top row: uniform noise  $\in [0, 0.05]$ . Middle row: uniform noise  $\in [0, 0.1]$ . Bottom row: uniform noise  $\in [0, 0.2]$ .

For the E-step of the EM algorithm we have to compute

$$(A.2) \quad p(\mu|b, x^{k-1}) \propto p(b|\mu, x^{k-1})p(\mu|x^{k-1}).$$

Using an improper prior for  $\mu$  we calculate a bivariate Gaussian representation for  $\mu$  based on reconstruction  $x^{k-1}$  and given projection values  $b$ . So we want to find

$$(A.3) \quad p(\mu|b, x^{k-1}) = \frac{1}{2\pi|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mu-\bar{\mu})\Sigma^{-1}(\mu-\bar{\mu})},$$

$$(A.4) \quad \Sigma = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_0 \\ \rho\sigma_0\sigma_1 & \sigma_0^2 \end{bmatrix},$$

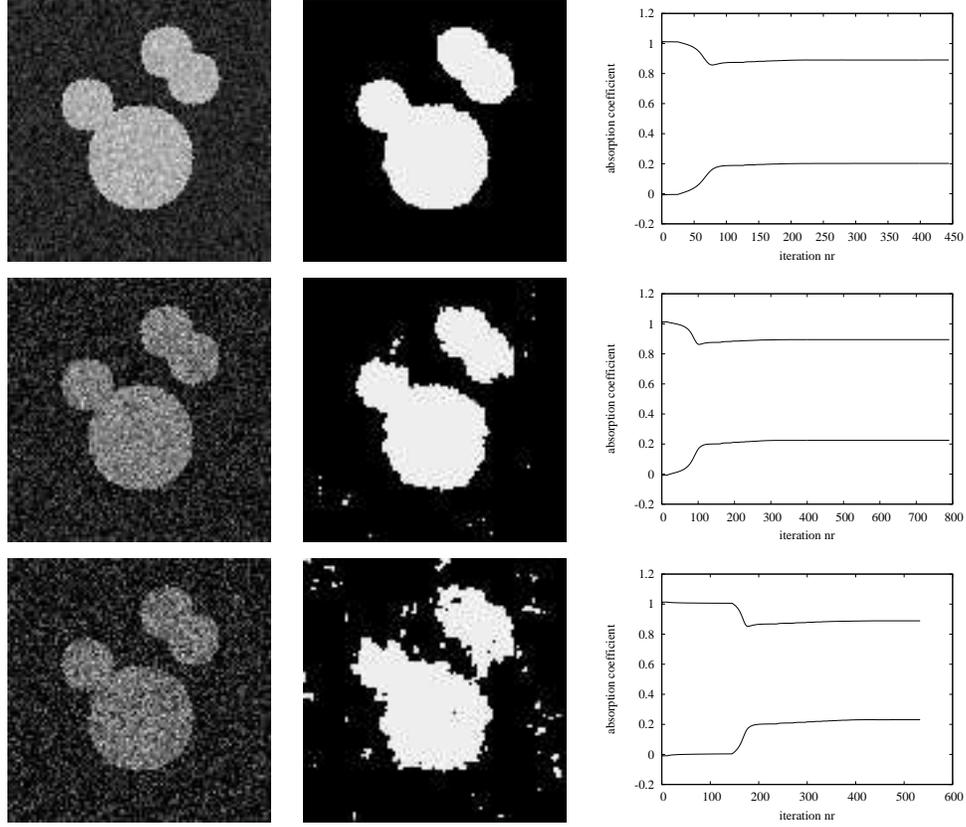


Fig. 6. Gaussian noise : Reconstruction result with Gaussian noise ( mean=0 ) for a  $100 \times 100$  image using 5 projections over 180 degree. Left column: Ground truth with noise and  $\mu_0 = 0.2$  and  $\mu_1 = 0.9$  Center column: Reconstruction result with adaptive DC-algorithm Right column: Developing of absorption coefficients. Top row: noise with  $\sigma = 0.01$  Middle row: noise with  $\sigma = 0.05$  Bottom row: noise with  $\sigma = 0.1$

with  $\mu = [\mu_1, \mu_0]^T$  and  $\rho := \text{cor}(\mu_0, \mu_1) = \frac{\sigma_{01}}{\sigma_0 \sigma_1}$  where  $\rho$  is the correlation of  $\mu_0$  and  $\mu_1$ . The covariance is given by

$$(A.5) \quad \text{cov}(\mu_0, \mu_1) = \rho \sigma_0 \sigma_1,$$

and we get the following moments

$$(A.6) \quad \langle \mu_i^2 \rangle = \mu_i^2 + \sigma_i^2,$$

$$(A.7) \quad \langle \mu_i \rangle = \mu_i,$$

$$(A.8) \quad \langle \mu_0 \mu_1 \rangle = \text{cov}(\mu_0, \mu_1) + \mu_0 \mu_1.$$

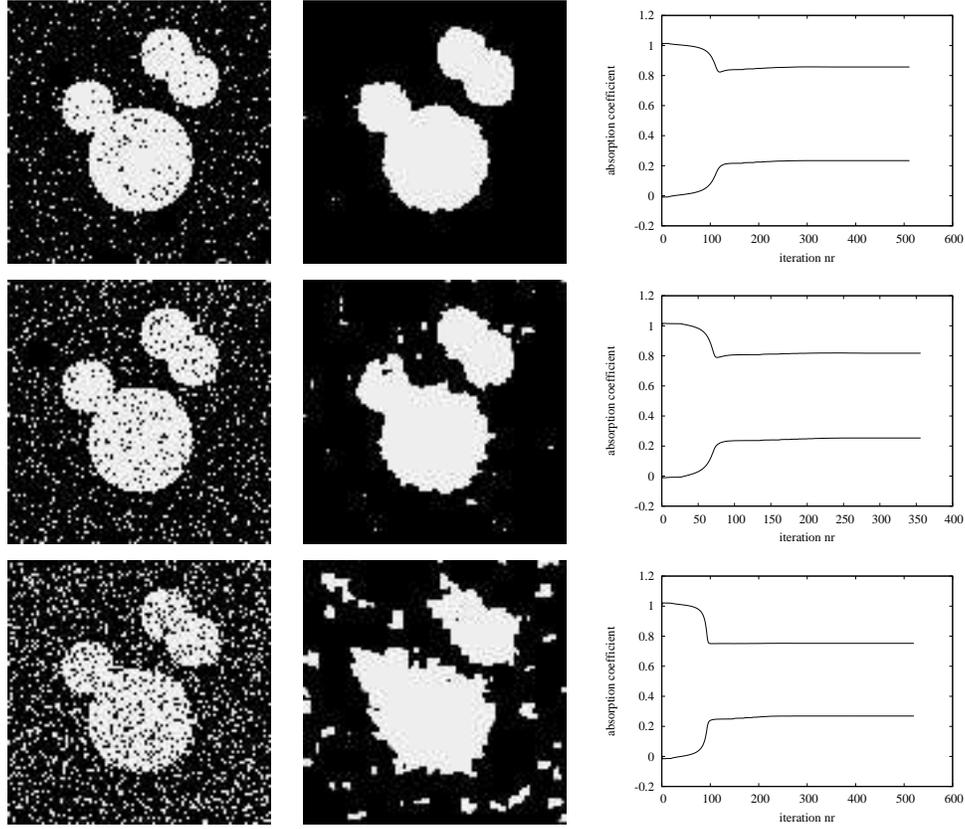


Fig. 7. Salt and pepper noise: Reconstruction result with salt and pepper noise for a  $100 \times 100$  image using 5 projections over 180 degree. Left column: Ground truth with noise and  $\mu_0 = 0.2$  and  $\mu_1 = 0.9$  Center column: Reconstruction result with adaptive DC-algorithm Right column: Developing of absorption coefficients. Top row: 5 % noise. Middle row: 10 % noise. Bottom row: 20 % noise.

## A.2 $\mu$ Estimation

To estimate the normal parameters of  $\mu$  we convert the quadratic functional

$$(A.9) \quad (b - (Ax\mu_1 + A(e-x)\mu_0))^T (b - (Ax\mu_1 + A(e-x)\mu_0))$$

into a Gaussian representation for  $\mu$

$$(A.10) \quad \left( \begin{pmatrix} \mu_1 \\ \mu_0 \end{pmatrix} - \begin{pmatrix} \tilde{\mu}_1 \\ \tilde{\mu}_0 \end{pmatrix} \right)^T \Sigma^{-1} \left( \begin{pmatrix} \mu_1 \\ \mu_0 \end{pmatrix} - \begin{pmatrix} \tilde{\mu}_1 \\ \tilde{\mu}_0 \end{pmatrix} \right).$$

We calculate  $\Sigma$  with

$$(A.11) \quad M := \begin{bmatrix} (Ax)^T Ax & (Ax)^T A(e-x) \\ (A(e-x))^T Ax & (A(e-x))^T A(e-x) \end{bmatrix},$$

$$(A.12) \quad \Sigma = M^{-1} = \begin{bmatrix} \gamma_1 & \gamma_2 \\ \gamma_2 & \gamma_3 \end{bmatrix},$$

which yields the estimates  $\hat{\sigma}_1 = \sqrt{\gamma_1}$ ,  $\hat{\sigma}_0 = \sqrt{\gamma_3}$  and  $\hat{\rho} = \frac{\gamma_2}{\hat{\sigma}_0 \hat{\sigma}_1}$ . Now the estimation  $\hat{\mu}$  is given by

$$(A.13) \quad \begin{pmatrix} \hat{\mu}_1 \\ \hat{\mu}_0 \end{pmatrix} = \Sigma \left( b^T \begin{bmatrix} Ax & A(e-x) \end{bmatrix} \right).$$

### A.3 Q function

Lets define

$$(A.14) \quad Q^* := \log p(x) + \int \log(p(b|\mu, x))p(\mu|x^{(k-1)}, b)d\mu.$$

Now we can rewrite the first integral term to

$$(A.15) \quad \log p(b|\mu, x) = -0.5(b - (Ax\mu_1 + A(e-x)\mu_0))^T$$

$$(A.16) \quad (b - (Ax\mu_1 + A(e-x)\mu_0))$$

$$(A.17) \quad = c_0(x, b)\mu_0^2 + c_1(x, b)\mu_0 + c_2(x, b)\mu_1^2$$

$$(A.18) \quad + c_3(x, b)\mu_1 + c_4(x, b)\mu_1\mu_2 + c_5(x, b),$$

with  $c_0 \dots c_5$  dependent on  $x$  and  $b$ . Note that  $c_0 \dots c_5$  can be calculated by writing  $\log p(b|\mu, x)$  in terms of  $\mu_0^2, \mu_0, \mu_1^2, \mu_1, \mu_1\mu_0$ . To keep notation simple, we will now use  $c_i$  instead of  $c_i(x, b)$ . For the E-step this yields

$$(A.19) \quad Q^* = \log p(x) + \int \log(p(b|\mu, x))p(\mu|x^{(k-1)}, b)d\mu$$

$$(A.20) \quad = \log p(x)$$

$$(A.21) \quad + \int (c_0\mu_0^2 + c_1\mu_0 + c_2\mu_1^2 + c_3\mu_1 + c_4\mu_1\mu_2 + c_5)p(\mu|x^{(k-1)}, b)d\mu.$$

As we use Gaussian representation of  $p(\mu|x^{(k-1)}, b)$ , the integration can be done analytically by using the moments of a normal distribution. This yields the following four cases for  $\mu_1$  based on the estimations  $\hat{\mu}_1, \hat{\mu}_0, \hat{\sigma}_1, \hat{\sigma}_0$  and  $\hat{\rho}$ :

$$(A.22) \quad \int_{\mu_1} \int_{\mu_0} \mu_1^2 p(\mu_0, \mu_1|x^{(k-1)}, b)d\mu_0 d\mu_1 = \hat{\mu}_1^2 + \hat{\sigma}_1^2,$$

$$(A.23) \quad \int_{\mu_1} \int_{\mu_0} \mu_1 p(\mu_0, \mu_1 | x^{(k-1)}, b) d\mu_0 d\mu_1 = \hat{\mu}_1,$$

$$(A.24) \quad \int_{\mu_1} \int_{\mu_0} \mu_1 \mu_0 p(\mu_0, \mu_1 | x^{(k-1)}, b) d\mu_0 d\mu_1 = \hat{\rho} \hat{\sigma}_1 \hat{\sigma}_0 + \hat{\mu}_1 \hat{\mu}_0,$$

$$(A.25) \quad \int_{\mu_1} \int_{\mu_0} p(\mu_0, \mu_1 | x^{(k-1)}, b) d\mu_0 d\mu_1 = 1.$$

The corresponding terms for  $\mu_0$  can be obtained accordingly. Given  $\hat{\mu}$ ,  $\hat{\sigma}$  and  $\hat{\rho}$  yields for  $Q^*$ :

$$(A.26) \quad Q^* := F^*(x; \hat{\mu}, \hat{\sigma}, \lambda) := \alpha(\langle x, Lx \rangle) + \lambda \langle x, e - x \rangle - (c_0(\hat{\mu}_1^2 + \hat{\sigma}_1^2) + c_1 \hat{\mu}_1$$

$$(A.27) \quad + c_2(\hat{\mu}_0^2 + \hat{\sigma}_0^2) + c_3(\hat{\mu}_0)$$

$$(A.28) \quad + c_4(\hat{\mu}_0 \hat{\mu}_1 + \hat{\rho} \hat{\sigma}_0 \hat{\sigma}_1) + c_5).$$