A Appendix

Dual-Energy Alternating Minimization (DEAM) Algorithm

To implement JSIR method we need the following equations and algorithm following Jingwei[46]. Solving the maximum log-likelihood estimation (MLE) problem

$$\min_{c} \sum_j \sum_y \left[Q_j(y,c) - d_j(y)\log Q_j(y,c)\right], \tag{50}$$

where

$$Q_j(y,c) = I_{0,j}(y) \sum_E \psi_j(y,E) e^{-\sum_i \mu_i(E) \sum_x h(y,x) c_i(x)}. \tag{51}$$

Solve the following instead of equation (50)

$$\min_{p,q} F(p,q) = \sum_y \sum_j \sum_E I(p_j(y,E)||q_j(y,E))$$

$$= \sum_y \sum_j \sum_E p_j(y,E)\log\frac{p_j(y,E)}{q_j(y,E)} - p_j(y,E) + q_j(y,E) \tag{52}$$

Such that

$$\sum_E p_j(y,E) = d_j(y)$$

$$q_j(y,E) = I_{0,j}(y)\psi_j(y,E)e^{-\sum_i \mu_i(E) \sum_x h(y,x) c_i(x)}.$$

Considering only the terms that contain p we have

$$\min_{p} F_p(p) = \sum_{y,j,E} \left[p_j(y,E)\log\frac{p_j(y,E)}{q_j(y,E)} - p_j(y,E) + q_j(y,E)\right], \tag{53}$$

Susch that

$$\sum_E p_j(y,E) = d_j(y)$$

The minimizer of problem (53) after applying Lagrange multiplier is

$$p_j(y,E) = \frac{q_j(y,E)}{\sum_E q_j(y,E)} d_j(y) \tag{54}$$

Since q is a function of c, minimizing with respect to c is equivalent to minimizing with respect to q; so, considering only terms that contains c

$$\min_{c} F_c(c) = \sum_{y,j,E} - p_j(y,E)\log q_j(y,E) + q_j(y,E)$$

$$= \sum_{y,j,E} \left[I_{0,j}(y)\psi_j(y,E)e^{-\sum_i \mu_i(E) \sum_x h(y,x) c_i(x) + p_j(y,E) \sum_i \mu_i(E) \sum_x h(y,x) c_i(x)}\right]. \tag{55}$$

After convex decomposition, we have

$$\hat{F}_c(c) = \sum_{y,j,E} [\hat{q}_j(y,E) \sum_{i,x} \frac{\mu_i(E)h(y,x)}{Z_i(x)} e^{-Z_i(x)(c_i(x)-\hat{c}_i(x))}$$

$$+p_j(y,E) \sum_i \mu_i(E) \sum_x h(y,x)c_i(x)] \quad (56)$$

where

$$\hat{q}_j(y,E) = I_j(y,E)e^{-\sum_{i,x}\mu_i(x)h(y,x)\hat{c}_i(x)} \quad (57)$$

$$Z_i(x) = \sum_{y,E} \mu_i(E)h(y,x). \quad (58)$$

$\hat{c}_i(x)$ is the current estimate of $c_i(x)$ and Z is a parameter that guarantees convergence. The surrogate function's derivative with respect to $c_i(x)$ is

$$\frac{\partial \hat{F}_c}{c_i(x)} = \sum_{y,j,E} [-\hat{q}_j(y,E)\mu_i(E)h(y,x)e^{-Z_i(x)(c_i(x)-\hat{c}_i(x))} + p_j(y,E)\mu_i(E)h(y,x)] \quad (59)$$

Solving equation(59) equal to zero, we have final $c_i(x)$ update to be

$$c_i(x) = \hat{c}_i(x) - \frac{1}{Z_i(x)}\log(\frac{\sum_{y,j,E} p_j(y,E)\mu_i(E)h(y,x)}{\sum_{y,j,E} q_j(y,E)\mu_i(E)h(y,x)}) \quad (60)$$

To suppress the noise, a neigborhood smoothing function can be added to the surrogate function for example Huber-type penalty of the form[33,47]

$$R(c) = \sum_i \sum_x \sum_{\bar{x}\in\aleph_x} w_{x\bar{x}}\phi(c_i(x) - c_i(\bar{(x)})), \quad (61)$$

where

$$\phi(t) = \frac{1}{\delta^2}(\delta|t| - \log(1+\delta|t|)) \quad (62)$$

$\aleph_x$ is the set of neighborhood of the pixel x, and the corresponding weights $w_{x\bar{x}}$ are equal to the inverse distance between pixels x and $\bar{x}$.

The regularized DEAM cost function for the term containing c is

$$\min_c \hat{F}_c(c) = \sum_{y,j,E} [\hat{q}_j(y,E) \sum_{i,x} \frac{\mu_i(E)h(y,x)}{Z_i(x)} e^{-Z_i(x)(c_i(x)-\hat{c}_i(x))}$$

$$+p_j(y,E) \sum_i \mu_i(E) \sum_x h(y,x)c_i(x)] + \lambda R(c). \quad (63)$$

The basis-material image pixels $c_i(x)$ are coupled in penalty term $\phi(c_i(x) - c_i(\bar{x}))$; so, after being decoupled by convex decomposition the resulting cost function is

$$\min_c \hat{F}_c(c) = \sum_{y,j,E} [\hat{q}_j(y,E) \sum_{i,x} \frac{\mu_i(E)h(y,x)}{Z_i(x)} e^{-Z_i(x)(c_i(x)-\hat{c}_i(x))}$$

$$+p_j(y,E) \sum_i \mu_i(E) \sum_x h(y,x)c_i(x)] + \lambda \sum_i \sum_x \sum_{\bar{x}\in\aleph_x} \phi(2c_i(x) - \hat{c}_i(x) - \hat{c}_i(\bar{x})). \quad (64)$$

The corresponding derivatives are:

$$\frac{\partial \hat{F}_c}{\partial c_i(x)} = \sum_{y,j,E} [-\hat{q}_j(y,E)\mu_i(E)h(y,x)e^{-Z_i(x)(c_i(x)-\hat{c}_i(x))+p_j(y,E)\mu_i(E)h(y,x)}] + \lambda \sum_{\bar{x}\in\aleph_x} w_{x\bar{x}} \frac{2c_i(x)-\hat{c}_i(x)-\hat{c}_i(\bar{x})}{1+\delta|2c_i(x)-\hat{c}_i(x)-\hat{c}_i(\bar{x})|} \quad (65)$$

and

$$\frac{\partial^2 \hat{F}_c}{\partial c_i(x)^2} = \sum_{y,j,E} \hat{q}_j(y,E)\mu_i(E)h(y,x)Z_i(x)e^{-Z_i(x)(c_i(x)-\hat{c}_i(x))} + \lambda \sum_{\bar{x}\in\aleph_x} w_{x\bar{x}} \frac{2}{(1+\delta|2c_i(x)-\hat{c}_i(x)-\hat{c}_i(\bar{x})|)^2} \quad (66)$$

DEAM Algorithm can be wriiten as

Initialize c_i, and compute the corresponding $\hat{q}_j(y,E)$ and $\hat{p}_j(y,E)$.

if without penalty then

for $n = 0$ to $N-1$ do

for each i do

Set $c_i(x)^{(n)}$ as the current estimate.

Update $\hat{q}_j(y,E)$ with (57)

Update $\hat{p}_j(y,E)$ with (54)

Update $c_i(x)^{(n+1)}$ with (60)

end

end

end

if with penalty then

for $n = 0$ to $N-1$ do

for each j do

Set $c_i(x)^{(n)}$

Update $c_i(x)^{(n+1)}$ with Newton's method, and the first order and second order

derivatives are from (65) and (66).

end

end

end

Line Integral Alternating Minimization (LIAM) Algorithm

The LIAM algorithm presented follows from Chen et al[48].

Write the log-likelihood term as

$$I(d_j||F_j) = \min_{p_j \in \ell} I(p_j||f_j) \tag{67}$$

$$\epsilon_j = \{f_j: f_j(y,E) = I_{0,j}(y,E)e^{-\sum_{i=1}^{2} L_i(y)\mu_i(E)}\} \tag{68}$$

$$\ell(d_j) = \{p_j(y,E) \geq 0: \sum_E p_j(y,E) = d_j(y)\} \tag{69}$$

$L_i(y) = \sum_i h(y,x)c_i(x),\ i \in \{1,2\}$

The I-divergence includes sums over source-detector pairs y and energies E,

$$I(p_j||f_j) = \sum_y \sum_E \left[p_j(y,E)\log\frac{p_j(y,E)}{f_j(y,E)} - p_j(y,E) - f_j(y,E)\right] \tag{70}$$

The reformulated problem becomes

$$\min_{c- \geq 0, f_j \in \epsilon_j} \min_{p_j \in \ell(d_j)} \sum_{j=1}^{2} I(p_j||f_j) + \beta \sum_{i=1}^{2} I(L_i||Hc_i). \tag{71}$$

The gradient for $(L_1(y), L_2(y))^T$ is $\nabla(y) = (\nabla_1(y), \nabla_2(y))^T$, where

$$\nabla_i(y) = \sum_{j=1}^{2} \sum_E \hat{p}_j(y,E)\mu_i(E)$$
$$- \sum_{j=1}^{2} \sum_E \mu_i(E) I_{0j} e^{-\sum_{i'=1}^{2} L_{i'}(y)\mu_{i'}(E)} + \beta\left(\log\frac{L_i(y)}{\sum_x h(y,x)\hat{c}_i(x)}\right) \tag{72}$$

The corresponding Hessian matrix is $\nabla^2(y) =$

$((\nabla_{11}^2(y), \nabla_{21}^2)^T, (\nabla_{12}^2(y), \nabla_{22}^2(y))^T)$, where

$$\nabla_{ii}^2(y) = \sum_{j=1}^2 \sum_E \mu_i^2(E) I_{0j} e^{-\sum_{i'=1}^2 L_{i'}(y)\mu_{i'}(E)} + \frac{\beta}{L_i(y)}, \quad (73)$$

$$\nabla_{12}^2(y) = \nabla_{21}^2 = \sum_{j=1}^2 \sum_E \mu_1(E)\mu_2(E) I_{0j} e^{-\sum_{i'=1}^2 L_{i'}(y)\mu_{i'}(E)}. \quad (74)$$

The LIAM algorithm is as follows:

1.Set $k = 0$. Initial $\hat{c}_i^{(0)}, \hat{L}_i^{(0)}$.

2. Update $\hat{p}_j^{(k)}(y, E)$ according to

$\hat{f}_j^{(k)}(y, E) = I_{0j}(y, E) e^{-\sum_{i=1}^2 \hat{L}_i^{(k)}(y)\mu_i(E)}$,

$\hat{p}_j^{(k)}(y, E) = d_j(y) \frac{\hat{f}_j^{(k)}(y,E)}{\sum_{E'} \hat{f}_j^{(k)}(y,E')}$.

3. Update $\hat{L}_i^{(k+1)}(y)$ using Newton's method.

(i) Set $m = 0$. Let $\hat{L}_{i-Newton}^{(m=0)}(y) = \hat{L}_i^{(k)}(y)$.

(ii) $\hat{L}_{i-Newton}^{(m+1)} = \hat{L}_{i-Newton}^{(m)} - [\nabla^2(y)]^{-1}\nabla(y)$, where $\nabla^2(y)$, $\nabla(y)$ are evaluated at

$\hat{L}_{i-Newton}^{(m)}(y)$

(iii) Iterate until convergence to obtain nonnegative

$$\hat{L}_i^{(k+1)}(y) = \max(0, \hat{L}_{i-Newton}^{(m+1)}(y))$$

4. Update $\hat{c}_i^{k+1}(x)$ using ID algorithm

(i) Set $n = 0$. Let $\hat{c}_{i-ID}^{(n=0)}(x) = \hat{c}_i^{(k)}(x)$.

(ii) $\hat{c}_{i-ID}^{(n+1)}(x) = \frac{\hat{c}_{i-ID}^{(n)}(x)}{\sum_y h(y,x)} \sum_y h(y,x) \frac{\hat{L}_i^{(k+1)}(y)}{\sum_x h(y,x)\hat{c}_{i-ID}^{(n)}(x)}$.

(iii) Iterate until convergence, $\hat{c}_i^{(k+1)}(x) = \hat{c}_{i-ID}^{(n+1)}(x)$.

5. Iterate 2 through 4 until convergence.

Step 4 for c_i update is same as ID algorithm update in Snyder et al[49]. If regularization term, $R(c)$ is added, the total objective function becomes

$$\min_{L_i,c_i\geq 0} \sum_{j=1}^{2} I(d_j||F_j) + \beta \sum_{i=1}^{2} [I(L_i||Hc_i) + \lambda R(c_i)]$$

$$= \min_{L_i,c_i\geq 0} \sum_{j=1}^{2} \sum_y [d_j(y)\log\frac{d_j(y)}{F_j(y)} - d_j(y) + F_j(y)]$$

$$+\beta \sum_{i=1}^{2} \sum_y [L_i(y)\log\frac{L_i(y)}{\sum_x h(y,x)c_i(x)} - L_i(y) + \sum_x h(y,x)c_i(x)]$$

$$+\beta\lambda \sum_{i=1}^{2} \sum_x \sum_{s\in\aleph_x} w_{x,s}\psi(c_i(x) - c_i(s)), \quad (75)$$

where λ is a scalar that reflects the amount of smoothness desired for fixed β, and $\beta\lambda$ controls the trade-off between data fit (I-divergence) and the image smoothness (regularization). After the update for $\hat{L}_i^{(k+1)}(y)$ (in step 3), the decoupled objective function for every $\hat{c}_i^{(k+1)}(x)$ is,

$$\min_{\hat{c}_i^{(k+1)}(x)\geq 0} f(\hat{c}_i^{(k+1)}) =$$

$$\sum_y [\hat{L}_i^{(k+1)}(y)\hat{p}_i^{(k+1)}(x,y)\log\frac{\hat{L}_i^{(k+1)}(y)\hat{p}_i^{(k+1)}(x,y)}{h(y,x)\hat{c}_i^{(k+1)}(x)} +$$

$$h(y,x)\hat{c}_i^{(k+1)}(x) + \lambda \sum_{s\in\aleph_x} w_{x,s}\psi(2\hat{c}_i^{(k+1)}(x) - \hat{c}_i^{(k)}(x) -$$

$$\hat{c}_s^{(k)})] \quad (76)$$

where $\hat{p}_i^{(k+1)}(x,y) = \frac{h(y,x)\hat{c}_i^{(k)}(x)}{\sum_{x'} h(y,x')\hat{c}_i^{(k)}(x')}$

Step 4 in algorithm for unregularized can be replace with trust region Newton's method in regularized LIAM as below:

1.Set $k = 0$. Initial $\hat{c}_i^{(0)}$,$\hat{L}_i^{(0)}$.

2. Update $\hat{p}_j^{(k)}(y,E)$ according to

$\hat{f}_j^{(k)}(y,E) = I_{0j}(y,E)e^{-\sum_{i=1}^{2}\hat{L}_i^{(k)}(y)\mu_i(E)}$,

$$\hat{p}_j^{(k)}(y,E) = d_j(y)\frac{\hat{f}_j^{(k)}(y,E)}{\sum_{E\prime}\hat{f}_j^{(k)}(y,E\prime)}.$$

3. Update $\hat{L}_i^{(k+1)}(y)$ using Newton's method.

(i). Set $m = 0$. Let $\hat{L}_{i-Newton}^{(m=0)}(y) = \hat{L}_i^{(k)}(y)$.

(ii). $\hat{L}_{i-Newton}^{(m+1)} = \hat{L}_{i-Newton}^{(m)} - [\nabla^2(y)]^{-1}\nabla(y)$, where $\nabla^2(y)$, $\nabla(y)$ are evaluated at

$$\hat{L}_{i-Newton}^{(m)}(y)$$

(iii). Iterate until convergence to obtain nonnegative

$$\hat{L}_i^{(k+1)}(y) = \max(0, \hat{L}_{i-Newton}^{(m+1)}(y))$$

4. Calculate $\hat{c}_{new}$ using the trust region Newton's method.

(i). Set $n = 0$, $0 < \eta_1 < \eta_2 < 1$, $0 < \gamma_1 < 1 < \gamma_2$.

Initialize trust region radius $\Delta^{(n=0)}$. Let $\hat{c}^{(n=0)} = \hat{c}_{old}$

(ii). Compute regular Newton's method update

$$\Delta\hat{c}^{(n+1)} = \max(\frac{-f'(\hat{c}^{(n)})}{f''(\hat{c}^{(n)})}, -\hat{c}^{(n)})$$

(iii).Calculate actual reduction

$f(\hat{c}^{(n)}) - f(\hat{c}^{(n)} + \Delta\hat{c}^{(n+1)})$.

(iv).Calculate predicted reduction

$-\Delta\hat{c}^{(n+1)}f'(\hat{c}_{old}) - \frac{1}{2}(\Delta\hat{c}^{(n+1)})^2 f''(\hat{c}_{old})$.

(v). Calculate the ratio

$$\rho^{(n+1)} = \frac{ActualReduction}{PredictedReduction}$$

(vi). If $\rho^{(n+1)} < \eta_1$,

set $\Delta^{(n+1)} = \gamma_1\Delta^{(n)}$, $\hat{c}^{(n+1)} = \hat{c}^{(n)}$.

Else If $\rho^{(n+1)} > \eta_2$,

set $\Delta^{(n+1)} = \gamma_2\Delta^{(n)}$, $\hat{c}^{(n+1)} = \hat{c}^{(n)} + \Delta\hat{c}^{(n+1)}$.

Else

set $\Delta^{(n+1)} = \Delta^{(n)}$, $\hat{c}^{(n+1)} = \hat{c}^{(n)} + \Delta\hat{c}^{(n+1)}$.

End

(vii). Iterate until convergence to obtain $\hat{c}_{new} = \hat{c}^{(n+1)}$.

5. Iterate 2 through 4 until convergence.

In step 4, $\hat{c}_{old}$ is used to denote $\hat{c}_i^{(k)}(x)$ and $\hat{c}_{new}$ is used to denote $\hat{c}_i^{(k+1)}(x)$. Step (ii) in 4 is to enforce non-negativity for the component images(though sometimes the non-negativity is not needed). The choices of η_1, η_2, γ_1, γ_2 and $\Delta^{(n=0)}$ are empirical.

46 **Lu J**. Multi-Dimensional Extension of the Alternating Minimization Algorithm in X-ray Computed Tomography. [cited 15 December 2025]. Available from: https://openscholarship.wustl.edu/eng_etds/477/

47 **Fessler JA**, Ficaro EP, Clinthorne NH, Lange K. Grouped-coordinate ascent algorithms for penalized-likelihood transmission image reconstruction. *IEEE Trans Med Imaging* 1997; **16**: 166-175 [PMID: 9101326 DOI: 10.1109/42.563662]

48 **Chen Y**, O'Sullivan JA, Politte DG, Evans JD, Han D, Whiting BR, Williamson JF. Line Integral Alternating Minimization Algorithm for Dual-Energy X-Ray CT Image Reconstruction. *IEEE Trans Med Imaging* 2016; **35**: 685-698 [PMID: 26469126 DOI: 10.1109/TMI.2015.2490658]

49 **Snyder D**, Schulz T, O'sullivan J. Deblurring subject to nonnegativity constraints. *IEEE Trans Signal Process* 1992; **40**: 1143-1150 [DOI: 10.1109/78.134477]