



Norwegian
Meteorological Institute
met.no

met.no report

no. 2/2012

Remote sensing

Correcting and quantifying radar data

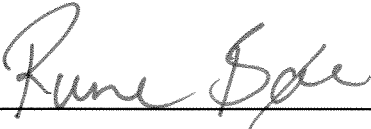
Christoffer A. Elo



Title Correcting and quantifying radar data	Date 2012-01-30
Section Remote sensing	Report no. no. 2/2012
Author(s) Christoffer A. Elo	Classification <input checked="" type="radio"/> Free <input type="radio"/> Restricted
	ISSN 1503-8025
	e-ISSN 1503-8025
Client(s) Energi Norge (EBL), The Research Council of Norway	NFR ES439901/193048
Abstract This report describes the filtering method for quality control and the estimation to rainfall products from radar reflectivity. The resulting radar products are used in data assimilation, hydrological modelling, meteorological forecasting and for the public. The project "Utnyttelse av værradardata i værvarslings- og tilsigsmodeller" has been a continuous project over the three last years, and is funded by EBL/EnergiNorge, NFR and met.no The report covers an innovative approach to sea clutter identification and a framework for processing the radar data to improve derived radar products. Another focus has been on estimating rainfall products from radar data, and how to adapt the range-independent correction method (VPR) for the Norwegian radars. Lastly, the report demonstrates the different products that are available for further use.	
Keywords quality control, radar data, reflectivity, VPR, clutter filtering, image processing	

Disiplinary signature

Responsible signature





Contents

1	Product chain	6
2	Processing	7
2.1	Processing in polar mesh	7
2.2	FoO – Frequency of Occurrence	7
2.2.1	Configuration	8
2.3	Sea clutter	8
2.3.1	The Fast Marching Method	9
2.3.2	Implementation and numerical results	9
2.3.3	Configuration	11
2.4	Sun flare	12
2.4.1	Configuration	13
2.5	Speckles	13
2.5.1	Configuration	13
2.6	Classification	14
3	Products	15
3.1	Polar to Cartesian coordinates	15
3.2	PPI – Plan Position Indicator	15
3.2.1	Configuration	15
3.3	CAPPI – Constant Altitude Plan Position Indicator	16
3.3.1	Configuration	16
3.4	SRI – Surface Rainfall Intensity	17
3.4.1	Configuration	17
3.5	Accumulation	19
4	Rain rate estimation	23
4.1	Examples	23
A	Radar propagation paths in ProRad	29
A.1	Functions	30
A.2	Interpolation	31
A.2.1	Functions	31

1 Product chain

The operational radar processing chain at met.no uses ProRad (available as GPL, Q2-2012) for processing radar data. There are currently eight weather radars in Norway, most of them located along the coast. These radars provides real-time weather observation used by the meteorologists, the industry, scientists and the public. In order to deliver radar data and products, the data goes through extensive quality control and data transformations, all handled by the ProRad system.

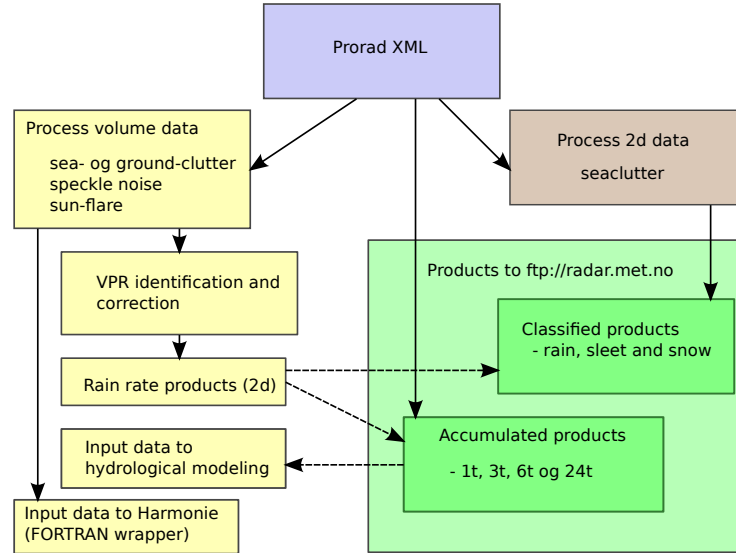


Figure 1: Product chain of ProRad.

The diagram above describe two parallell lines in the processing chain, volume data and 2d-data. Rainbow generates the 2d products before they are imported into the production chain by ProRad, where the data is flagged for sea clutter and the phase of precipitation, which the latter is decided by using data from numerical weather prediction model. Resulting products are used in rain rate accumulations, visualisation (Diana, met.no) and verification.

Volume data is the raw data that contains the full data set from Rainbow, these datasets have only been processed by the signal processor where e.g the Doppler filter has removed static noise. The data is placed in a polar mesh with range to 240.0km and discretized into 960 bins where each bin has a size of 0.25 km. Processing the raw data is useful for several things, firstly the data is not corrupted by converting to Cartesian coordinates, and secondly the spatial resolution is intact from the signal processor, i.e 0.25 km for bin size compared to 1.0km that are received from Rainbow. The most important reason for using the raw data is that the data consists of elevations that gives the vertical variability of the precipitation, this is useful for estimating rain rate and to identify clutter.

The polar volume data goes through extensive processing before the data is corrected for range-dependent error and converted into rain rate products or used as input for a numerical weather prediction model (NWP) or hydrological modelling. This insures that the delivered product is virtually free from clutter. Today, processed polar volume products are sent pre-operationally to the HARMONIE model as observation input to the data assimilation[13, 12], while derived 2d-products are distributed to `ftp://radar.met.no` and used in hydrological modelling, this includes 3h, 6h, 12h and 24h accumulated products and precipitation phase products. The delivered 2d products are in UTM-32 projection, and consists of a mosaic with the radars Rissa, Stad, Bømblo, Hegebostad and Hurum.

2 Processing

The radar reflectivity data are often corrupted by ground and sea clutter, anomalous propagation, birds and other non-meteorological echos. Ground clutter is usually easily removed with Doppler filter, i.e. reflectivity with radial velocity close to zero. The clutter generated by the sea waves are more difficult to identify, since it has similar velocities as precipitation. Thus, other kind of methods are needed to separate residual sea clutter from precipitation. There are several proposals in the literature, mostly based on statistical approaches, such as fuzzy logic[9] and neural networking[14]. Other methods analyse texture features such as in the DMI report [6] by Gill or make use of other datatypes and sources such as satellite data described in [3].

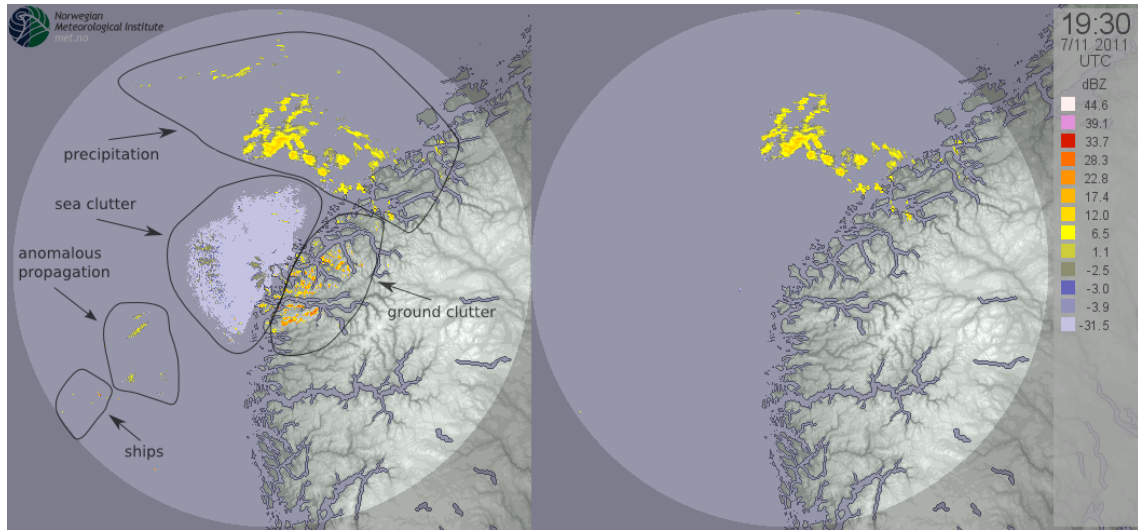


Figure 2: An example on various non-meteorological echos.

2.1 Processing in polar mesh

ProRad receives the raw data that are presented in polar coordinates, which is natural due to the scanning of the atmosphere. The radars in southern Norway scans 12 elevations, ranging non-equidistant from 0.5° to 15.0° . The radial spatial resolution for reflectivity is set to 0.25 km and has a range that goes out to 240.0 km, i.e. 960 data bins in each azimuth ray. The radar scans 360.0° azimuth, with a 1.0° fixed angle step, this results in an increasing scanned volume with respect to distance. At longer ranges, the measurement can be underestimated since the possibility of partially filled volume is increased. ProRad stores the data in an XML formatted file and announces that the dataset is ready for further processing.

The volume processing picks up the message and starts to process the product, starting from the highest elevation and looping through the slices. Each elevation is processed according to an XML configuration that describes which filters are activated. Each filter has a set of parameters that is dependent on elevation and radar site, this gives the flexibility to fine tune the filters for special needs. E.g. radar Stad has more sea and ground clutter than Hegebostad, and therefore needs a more carefully adjusted parameter setting.

The next subsections will describe each filter method and give an explanation on how the XML configuration is set up.

2.2 FoO – Frequency of Occurrence

Frequency of occurrence is a useful tool to determine where time-independent clutter is located. The FoO information is stored in binary files and needs to be generated based on seasons. The information

in the file is given in percent, such that the clutter can be identified by a simple threshold argument. Because of the varying radar propagation based on winter and summer time, the need to re-generate the files at the beginning of each season is important to use this information properly.

The files contains a buffer where each pixel represent the frequency of occurrence for a range of dates that are determined when the file was generated. The occurrence can be limited by giving a dBZ threshold. Generating the FoO files are easily done by calling the following command

Listing 1: Generate FoO files from command-line

```
$ prorad_generate_foo --storage-dir <path> --dbz-thres <dbz-value>
--slices <number-of-slices> <input-files>
```

It is also important to use data sets that have not been Doppler filtered by the signal processor, since the location of high occurrence pixels are often removed by the Doppler filter. The residual ground clutter from the doppler signal processor can be identified by the FoO map as a percent of probability.

The program will generate a static map of frequency of occurrence saved in a binary format given by the <path> argument. The second argument <dbz-value> thresholds the input files and only count values higher than <dbz-value>, this is more accurate for ground clutter if e.g the value is set to zero. The third argument <number-of-slices> is given if the user wants to reduced the number of slices. <input-files> are Rainbow5 XML files. Typically the files are taken from one or two months in the current season, and files from the previous year in the beginning of the seasons.

2.2.1 Configuration

The following configuration gives an example where 80.0% of the occurrences are identified as ground clutter. The second configuration option <foo-vpr> is used by the VPR method to exclude a larger range of lower occurrence, since higher quality, 50.0% in this example, is needed to generate vertical profiles but less important for e.g visualisation. Note that this filter can remove real precipitation.

Listing 2: XML configuration for foo clutter filtering

```
<foo-clutter-filter module="enabled">
  <foo_ground>80.0</foo_ground>
  <foo_vpr>50.0</foo_vpr>
</foo-clutter-filter>
```

2.3 Sea clutter

Identifying sea clutter has proven to be a difficult task. This is mainly because of the high horizontal variability and strong echos. The radar community has suggested several promising methods, but most of the people in the community are waiting for dual polarization to tackle the problem. Dual polarization can discriminate different types of precipitation, and therefore also give a falsification on sea clutter. In Norway (2012), only one of nine radars that can take advantage of dual polarisation, hence we need other methods to be able to distinguish between clutter and precipitation.

The goal of separating data into regions has been an ongoing development in computer graphics for some decades. An overview of several basic and well-known segmenting methods can be found in the book on image processing[8], here the author discusses segmentation methods based on statistics, edge detection and region growing. These techniques can be hard to implement, due to heavy data mining and parameter training that can be very time consuming depending on the scanning strategy and radar location.

A more sophisticated method is the level set method that uses curve evolution based on a numerical minimization problem for tracking interfaces and shapes. The drawback for this method is the speed, since it allows the curve to go either forward or backward, thus easily getting trapped in a piecewise continuous image.

Sethian proposed the Fast Marching Method[17], which is a stationary approach, i.e the region propagates only outwards and the arrival time for the problem is computed. We have implemented this popular method to segment our raw reflectivity data, and this has proven to be a fast and precise method, and easy to implement.

2.3.1 The Fast Marching Method

The Fast Marching Method is an approach for finding the evolution of a front, in our case, the front is the boundary of the clutter region. The method is using the radar reflectivity as input, and then propagates in the normal direction towards the boundary of the clutter region from a given initial seed point. If the radar site is chosen as the seed point, then the fastest way to the clutter boundary is one solution, as this is the minimum energy. It turns out that this “path” describes the clutter near the radar, hence segmenting the precipitation from the clutter. Note that this does not depend on the variance between pixels, but the minimum energy, also called the shortest distance. The evolution of the front is described by the Eikonal equation for a point $\mathbf{x} \in \mathbb{R}^2$ and the speed $F(\mathbf{x})$ of the front

$$\|\nabla T(\mathbf{x})\|F(\mathbf{x}) = 1, \tag{1}$$

where $T(\mathbf{x})$ is the arrival time and $\|\mathbf{x}\| = \sqrt{y_1^2 + y_2^2}$ is the Euclidean norm. If the speed function $F(\mathbf{x})$ is equal to one everywhere, then the solution is simply the signed function $T(\mathbf{x}) = |\mathbf{x}|$, i.e the front has equal speed in all direction and the solution is circular.

2.3.2 Implementation and numerical results

The results are obtained by giving the algorithm multiple seed points with low reflectivity near the radar. Each pixel that the algorithm passes is then assigned a flag that sets a bit value for sea clutter and an integer(0-100) value for percent of clutter probability (example in figure 4). The set of flags is then used as a quality check for further use of the radar reflectivity data. Other clutter types such as ground and sun-flare are also flagged.

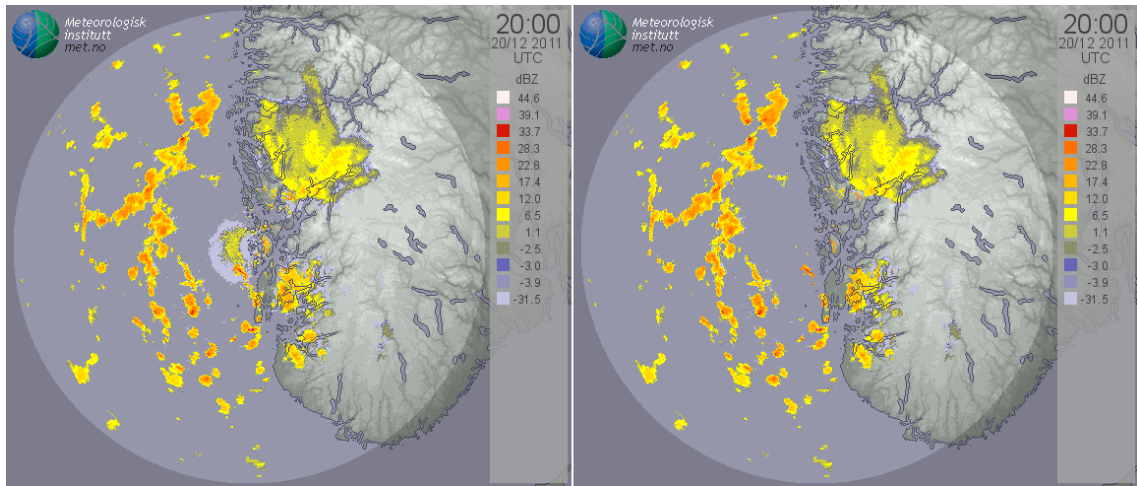


Figure 3: An example of radar Bømblo with sea clutter combined with precipitation. The image to the right is the result where sea clutter infected pixels are removed.

Figure 3 on page 9 are processed with the Fast Marching Method to identify sea clutter, no other processing has been done to mark other clutter. The image on the left in the figure shows the radar site at Bømblo, and the right image shows the same data, except that pixels flagged as clutter have been removed. Thus, illustrating how the method has propagated from the radar site and covered the sea clutter region. The figure also illustrates that the method does not affect the precipitation echos.

Algorithm 1: Fast marching method.

```
// Initialization
foreach pixel  $\in$  seed do
  Put pixel in freeze;
  foreach neighbour v  $\in$  frozen pixel do
    Compute value  $dist(v) = eikonal\_equation(mat)$  for v;
    if v  $\in$  heap then
      replace value  $dist(v)$  in heap
    else
      Put ( $dist(v)$ ) in heap
    end
  end
end
// Evolve
while heap is not empty do
  Extract v from heap;
  Put v in freeze;
  foreach neighbour vn  $\in$  v do
    if vn  $\notin$  freeze then
      Compute value  $dist(vn) = eikonal\_equation(mat)$  for v;
      if vn  $\in$  heap then
        replace value  $dist(vn)$  in heap
      else
        insert  $dist(vn)$  into heap
      end
    end
  end
end
end
```

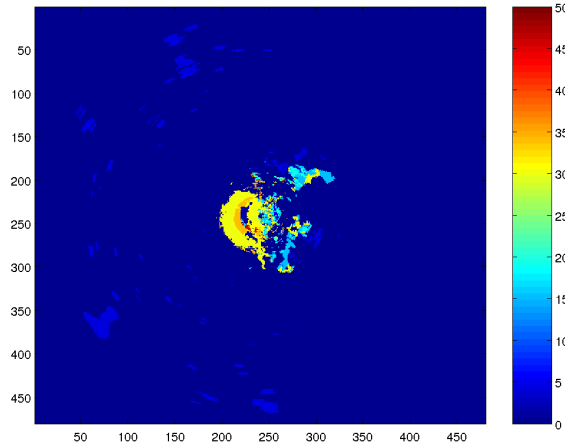


Figure 4: Probability for clutter, where the color bar is indicating probability (0-100).

Running the fast marching method is easily done by calling the function `fmm_evolve` with a few structures that needs to be initialized first, the algorithm itself are psuedo-coded in algorithm 1 on page 10.

Listing 3: `fmm_evolve` prototype in `fmm.h`

```
void fmm_evolve( const fmm_info_t fmm_info, const seed_t *seed, const matrix_t
    *mat,
    matrix_t *dist, matrix_uint8_t *narrow_band, matrix_uint8_t *freeze);
```

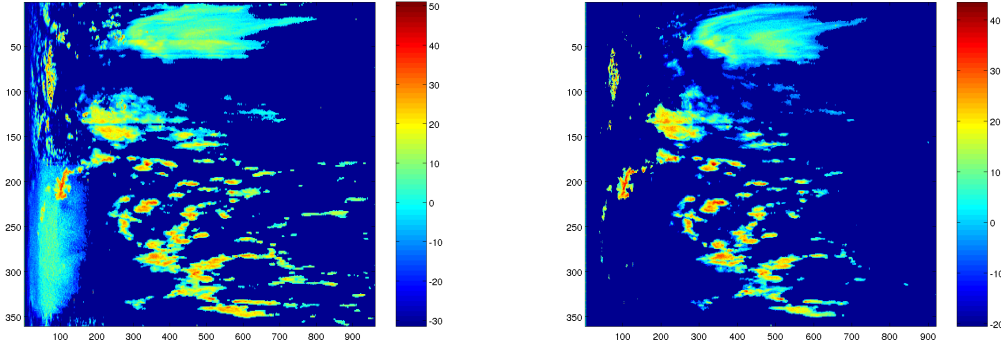
The first structure `fmm_info_t` contains two options: (i) `fmm_info_t.max_dist_value` that describes the time arrival $T(\mathbf{x})$ in equation (1), i.e the stopping criteria. The best value for this option is found by trail-and-error. The next option (ii) `fmm_info_t.speed_func` is a pointer to the speed function $F(\mathbf{x})$ in equation (1), this just the function $F(\mathbf{g}(\mathbf{x})) = 1/\mathbf{g}(\mathbf{x})$ for minimum energy where $\mathbf{g}(\mathbf{x})$ is the input image (figure 5(a)) given in the matrix structure `matrix_t *mat`. The output structures are defined in the last three arguments `matrix_t *dist`, `matrix_uint8_t *narrow_band`, `matrix_uint8_t *freeze` where `matrix_t *dist` holds the distance function $T(\mathbf{x})$ shown in figure 5(c). Figure 5(b) shows the clean dataset, where pixels identified from figure 5(c) are removed.

2.3.3 Configuration

Listing 4 shows an XML configuration for the sea clutter filtering method. The first option in this example, `<seed-min-thres>-8.0</seed-min-thres>`, sets the threshold for maximum dBZ value that are allowed for seed points, all pixels with dBZ value below this threshold is marked as seed points. The next option describes how far from the radar site the seed point algorithm are allowed to search. In this case (listing 4) all values under -8.0 dBZ within 50.0 km should be given as seed points to the FMM algorithm. The important stopping criteria `<fmm-max-dist-value>` is a value that is found by trial-and-error, and this example the value 4.3 is chosen high since the sea clutter near the Bømblo radar is much stronger than at any of the other sites. The figure 5(c) shows how the forward marching in the algorithm has stopped at 4.3.

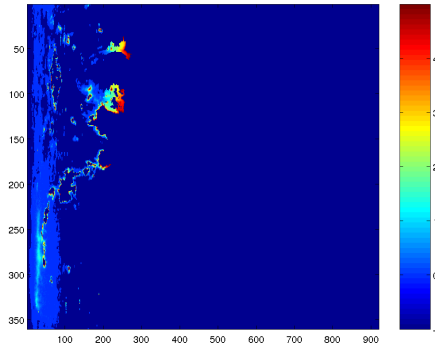
The last option `<fmm-otherclutter>` indicates if the method is constrained to data over sea, which is decided by a sea and land mask. If the argument is 1 the method will march at pixels over ground and flag reflectivity data as *other clutter*, this is useful for inland radars such as Hegebostad, where a lot of the clutter near the radar is caused by ground clutter and clear-air return. Figure 6(a) and the clean result in figure 6(b) is an example where pixels are affected by various clutter.

Listing 4: XML configuration for sea clutter filtering. First polar slice at the Bømblo site



(a) Original sample

(b) Cleaned with FMM



(c) The resulting distance function $T(\mathbf{x})$

Figure 5: The same dataset as in figure 3 in polar coordinates.

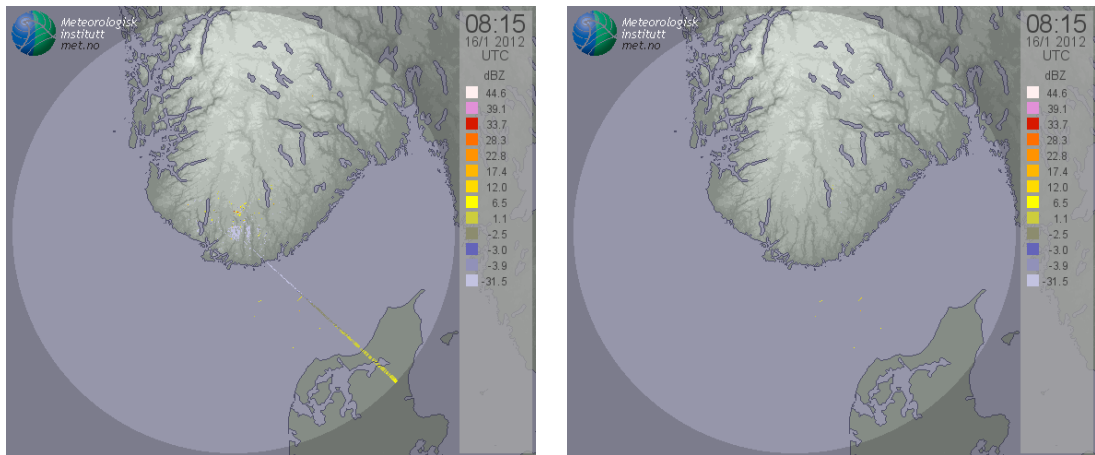
```
<sea-clutter-filter module="enabled">
  <seed-min-thres>-8.0</seed-min-thres>
  <seed-range>50.0</seed-range>
  <fmm-max-dist-value>4.3</fmm-max-dist-value>
  <fmm-otherclutter>0</fmm-otherclutter>
</sea-clutter-filter>
```

2.4 Sun flare

Sun flare contains microwave radiation that matches the wavelength of the radar, and the sensitive receiver in the radar will detect the sun flare. The straight ray is easily visually detected, but for accumulation, data assimilation and other use, can mistake it for precipitation. This particular clutter affects all the elevations, but often seen in 2d-products during sunrise and sunset that has not been quality controlled. The sunlight affects one or two rays in the polar volume, and the line segment is varying highly with respect to the dBZ value, this can be exploited by calculating the total variation (TV) for each line segment

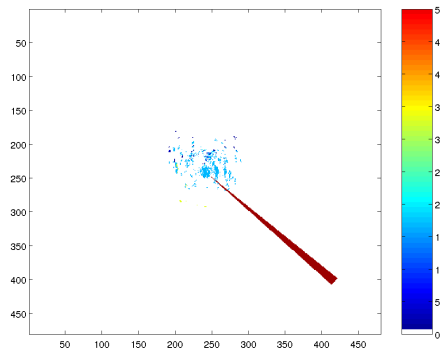
$$TV = \sum_{k=1}^{k < bins} |f(k) - f(k-1)|, \quad (2)$$

and set a threshold for the TV value. When the TV value is over the given threshold, each pixel in the line segment is flagged as “other-clutter” if the azimuth neighbours are zero. The sun flare in figure 6(a) removed in figure 6(b) by this method and the last image shows the probability of clutter, where the sun flare clutter has high probability while the clear-air return and ground clutter has low probability.



(a) Lowest PPI elevation

(b) Sun flare and ground clutter removed. Ground clutter is identified by FMM and static map



(c) Probability function

Figure 6: An example of sun flare at the Hegebostad radar.

2.4.1 Configuration

The configuration holds two arguments for two different norms to calculate if a line segment is affected by sun flare. The first option `tv-ratio-thres` refers to equation (2) and the second option is simply a $L2$ norm.

Listing 5: XML configuration for sun flare filtering

```
<sun-flare-filter module="enabled">
  <tv-ratio-thres>7.0</tv-ratio-thres>
  <l2-ratio-thres>75.0</l2-ratio-thres>
</sun-flare-filter>
```

2.5 Speckles

This filter removes speckle noise such as ships traffic and other small clusters in the data.

2.5.1 Configuration

The option `<cluster-count-thres>` is a threshold of how many pixels a cluster can hold before flagged as other-clutter. In this case all cluster sizes under 4 pixels are flagged.

Listing 6: XML configuration for speckle filtering

```
<speckle-filter module="enabled">
  <cluster-count-thres>4</cluster-count-thres>
</speckle-filter>
```

2.6 Classification

The classification of precipitation is obtained by calculating the “probability of rain” [10] equation, and the work is based on a previous article [7] by Gjertsen et. al. A new method that interpolates gridded model data onto the radar grid has been implemented in C++ and integrated into ProRad for cartesian products such as CAPPI and PPI. The method uses the 2 m temperature and humidity from numerical model (in this case UM 4 km), where the grid from the numerical model is interpolated to the radar spatial resolution, i.e 1. km for cartesian products. The classes of precipitation, snow, sleet and rain, are separated by setting a threshold for the obtained probability value. Figure 7 shows a mosaic of the southern radars in Norway, red colour scale for snow, yellow colour scale for sleet and blue colour scale for rain.

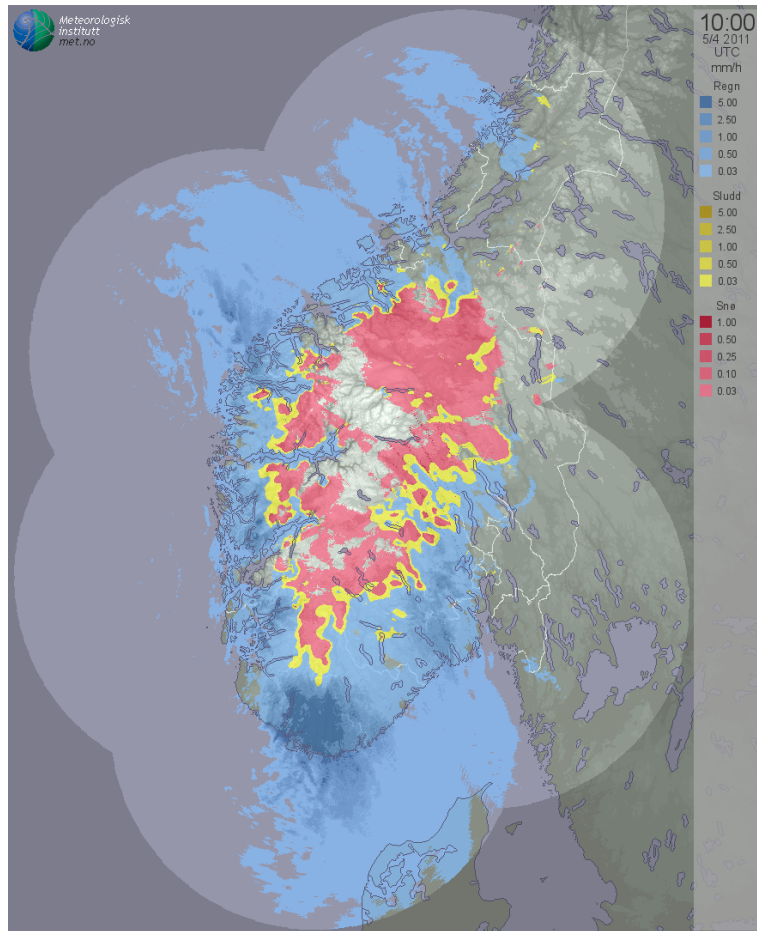


Figure 7: An example on phase classification of a mosaic consisting of the southern radars in Norway.

3 Products

The next step in the production chain, after the processing with all the methods for flagging clutter and noisy data, is to output products according to the user requirements. These are the final products that will be used for hydrological modelling, NWP and visualisation. After processing, ProRad announces that the filtered volume is ready, and this triggers the modules that generates different products according to a given XML configuration.

3.1 Polar to Cartesian coordinates

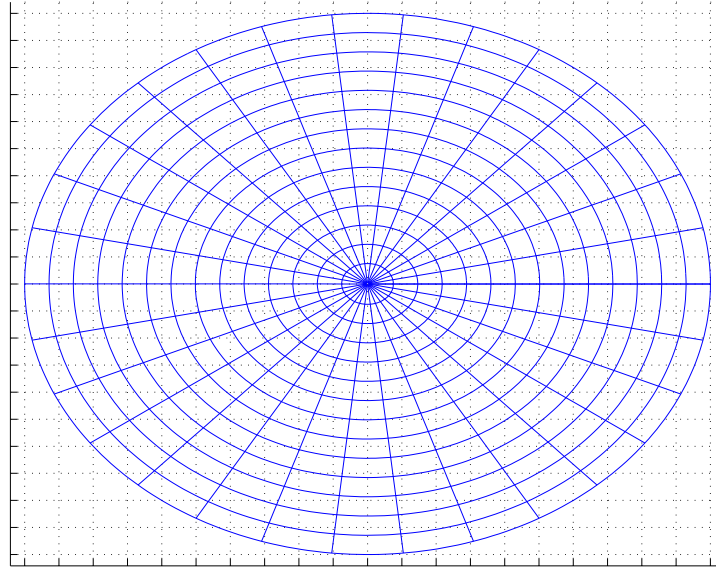


Figure 8: Polar and Cartesian grid for radar data.

The raw volume data are organised in polar coordinates, and these coordinates needs to be transformed into Cartesian coordinates for visualisation. This is done by the aid of inverse polar to Cartesian algorithm, looping through all the Cartesian pixels and finding the corresponding polar coordinates with trigonometric functions. A calculation based on bilinear interpolation with respect to range and azimuth is then used to find the corresponding polar dBZ data and flags.

In free space (no atmosphere) the radar beam would propagate as a straight line. However, the center of the beam is not following a straight line, but propagates as a curved line that is slightly bent towards the surface of the earth. A simple correction for this is to assume “normal atmosphere” and set the curvature to a constant value, normally $4/3$ of the actual curvature of earth.

3.2 PPI – Plan Position Indicator

A PPI is an elevation from a polar volume displayed as a Cartesian grid, like figure 8, and a PPI of the lowest elevation from the polar volume is often used to visualise precipitation near to the ground. The drawback of using the lower elevations as PPI is that they are often corrupted by sea and ground clutter, even after extensive quality control. The thin arc west for the radar in the processed product in figure 10(b) is residual clutter that the sea clutter method failed to identify.

3.2.1 Configuration

The following configuration example generates a PPI product with spatial resolution 1.0 km for both x and y direction. The number of pixel is set to 480×480 and only the first elevation is generated.

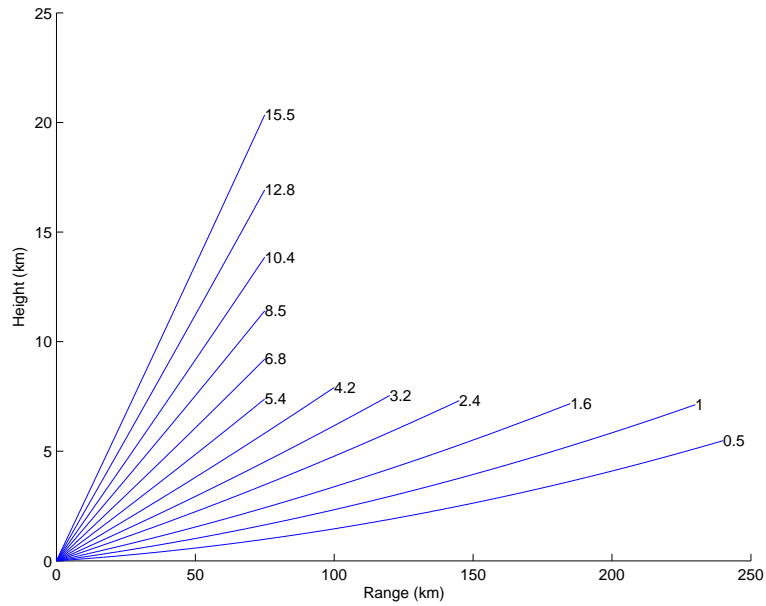


Figure 9: A scan strategy that illustrates the height of the beam center with respect to range after the curvature correction. The elevation angles and scan range for each elevation, are according to radar Stad scanning strategy.

Listing 7: XML configuration for the lowest PPI product

```
<ppi module="enabled">
  <slice no="0">
    <x-size>480</x-size>
    <y-size>480</y-size>
    <dx>1.0</dx>
    <dy>1.0</dy>
  </slice>
</ppi>
```

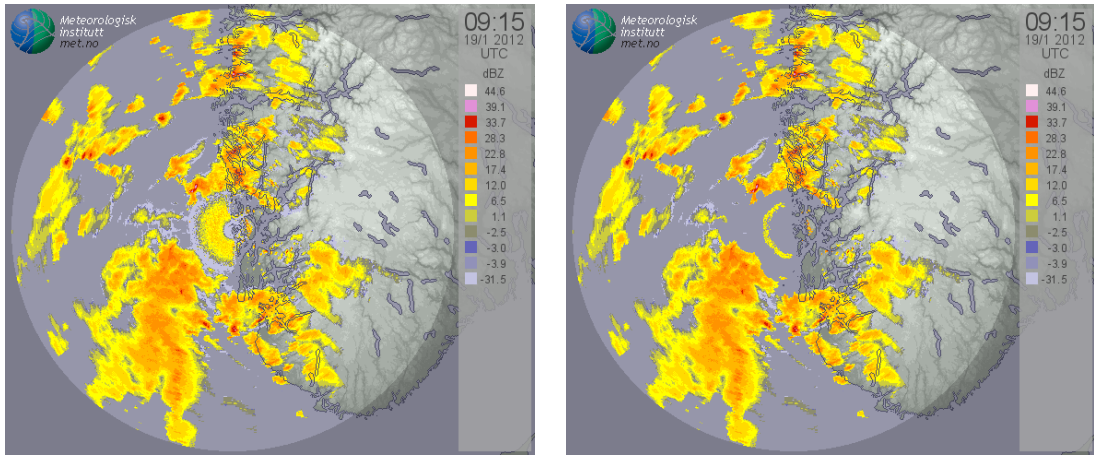
3.3 CAPPI – Constant Altitude Plan Position Indicator

One way to solve the residual sea clutter from the processing near the radar is to use CAPPI levels, this means a horizontal cross-section of constant height above the radar, taken from the volume, i.e higher elevations near the radar and avoiding corrupted data. However, choosing a constant height limits the range when the lowest elevation are higher than the constant level, e.g a CAPPI level of 1.5 km above mean sea level only goes out to ca. 150.0 km in range. If the CAPPI product uses the lowest PPI after the limited range as the examples in figure 11, the product is called “Pseudo-CAPPI”, or PCAPPI. A CAPPI product with a constant height level at e.g 1.5 km will give a product with much less clutter before processing, this is illustrated in 11(a) where the low reflectivity is residual clutter from the lower elevations. Making a horizontal cross-section of the filtered volume data results in a product that is almost completely free of clutter compared to the filtered PPI product in figure 10(b).

3.3.1 Configuration

This CAPPI product has the same spatial resolution as the PPI configuration in section 3.2.1. The cappi level is set to 1.5km above mean sea level and the last option <pseudo-cappi> indicates that this is a *Pseudo-CAPPI* product.

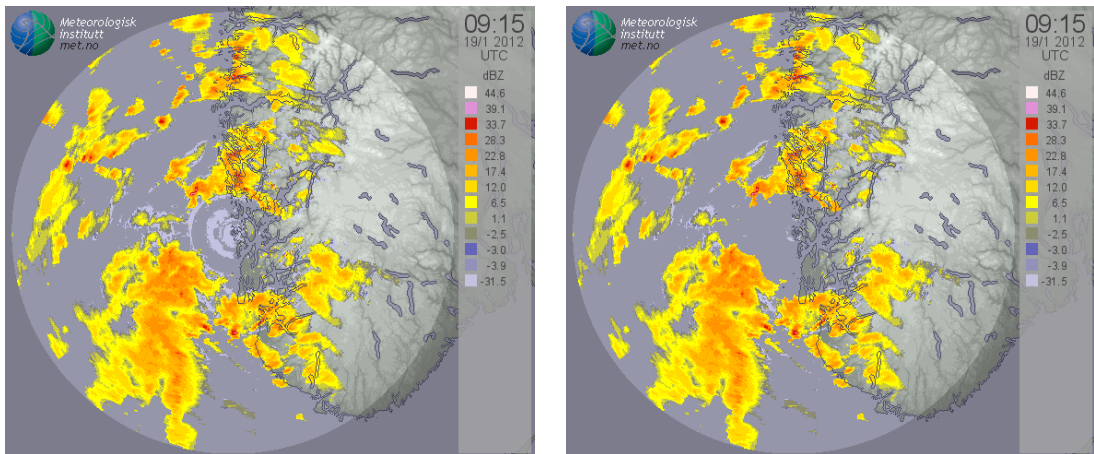
Listing 8: XML configuration for CAPPI products



(a) Affected by strong clutter

(b) After quality control

Figure 10: Lowest PPI product from the Bømblo radar.



(a) Small rings of sea clutter

(b) After processing

Figure 11: PCAPPI product, cappi height at 1.5km.

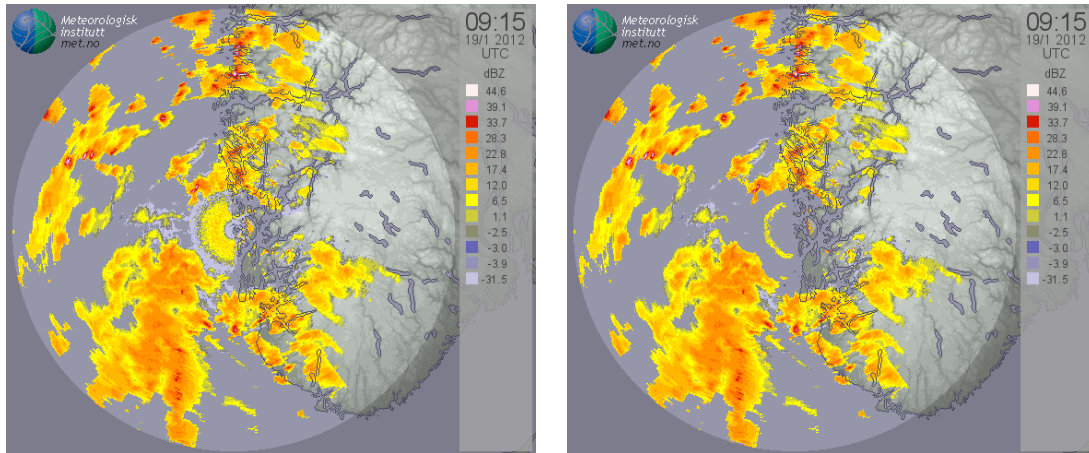
```
<cappi module="enabled">
  <x-size>480</x-size>
  <y-size>480</y-size>
  <dx>1.0</dx>
  <dy>1.0</dy>
  <cappi-level>1.5</cappi-level>
  <pseudo-cappi>1</pseudo-cappi>
</cappi>
```

3.4 SRI – Surface Rainfall Intensity

This is a product that uses the lowest PPI and project the aloft reflectivity data down to a reference height near to the ground 1.0km. The projection method is the so-called vertical profile of reflectivity correction (VPR), more details and examples can be found in section 4.

3.4.1 Configuration

The main ingredient for SRI products is the VPR corrected volume data, and the following configuration sets up the parameters for this method that is called by the SRI product generator. The



(a) Improved reflectivity

(b) After processing

Figure 12: SRI product, VPR reference height at 1.0km.

first parameter `<h-size>` describes the dimension of the height, together with the spatial resolution parameter `<dz>` given in km, which in this case is set to 0.2 km, results in a profile that has height equal to 10.0 km. The rest of the parameters describe Upper Bound (UB) and Lower Bound (LB) thresholds with respect to *height*(km), *range*(km) and *dBZ* values.

Listing 9: XML configuration for VPR correction

```
<vpr module="enabled">
  <h-size>50</h-size>
  <h-UB>10.0</h-UB>
  <h-LB>0.0</h-LB>
  <r-UB>120.0</r-UB>
  <r-LB>1.0</r-LB>
  <dbz-UB>50.0</dbz-UB>
  <dbz-LB>-10.0</dbz-LB>
  <dz>0.2</dz>
  <dr>0.25</dr>
</vpr>
```

After the profile is calculated, ProRad generates a SRI product with the following configuration. The resolution is the same as the PPI and CAPPI, and the last option `<h-ref>` is a height reference parameter for the VPR correction.

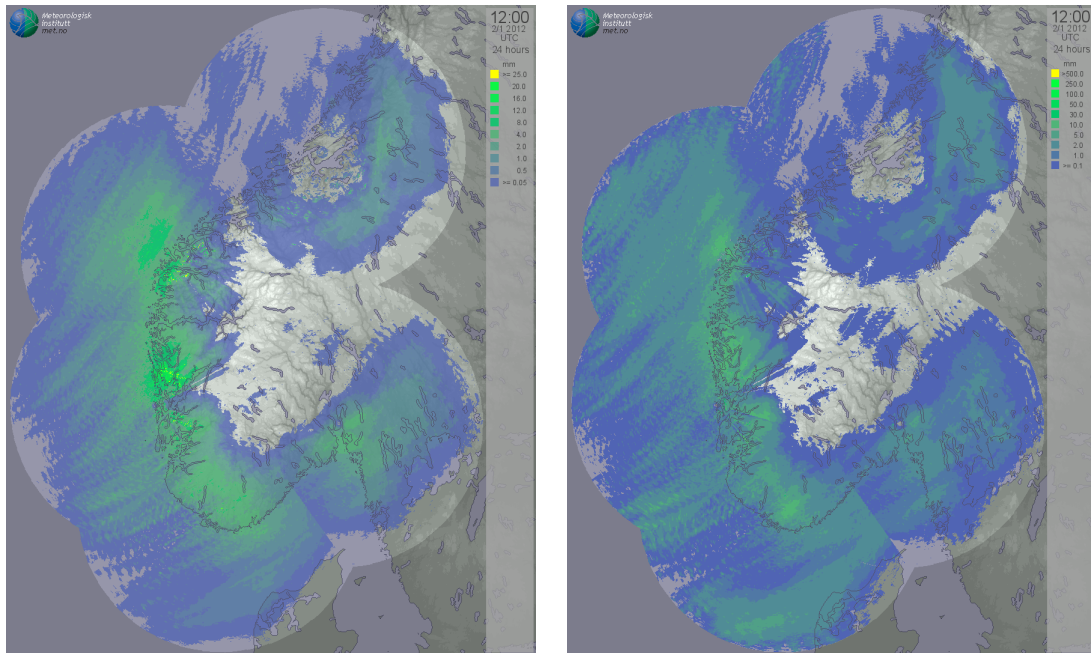
Listing 10: XML configuration for SRI products

```
<sri module="enabled">
  <x-size>480</x-size>
  <y-size>480</y-size>
  <dx>1.0</dx>
  <dy>1.0</dy>
  <h-ref>1.0</h-ref>
</sri>
```

3.5 Accumulation

Radar rainfall accumulation products are available at `ftp://ftp.met.no/projects/radar/radar/`, derived from PCAPPI(example figure 13(a)) and SRI(figure 13(b)) products. These products are georeferenced GeoTIFF images in UTM32-projection, and represent the total estimated rainfall over a particular time period, this case in 1 hour, 3 hours, 12 hours and 24 hours.

These products are useful for hydrologists for accurate inflow forecasting and for meteorologists in weather forecasting, the products are also important in flood warnings.



(a) PCAPPI mosaic product, cappi height at 1.5 km (b) VPR corrected SRI mosaic product, VPR reference height at 1.0 km

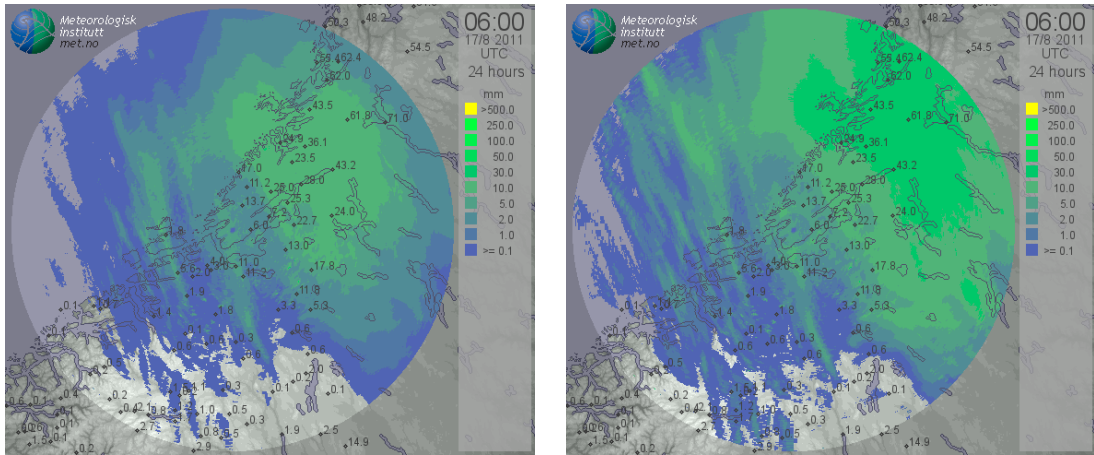
Figure 13: Mosaic product of the southern radars in Norway.

Figure 13(a) shows an enhanced total rainfall near radar Bømblo and radar Stad, and this is typically a bright band effect that can be mistaken for heavy precipitation. The image to the right in the figure shows the same situation, but corrected with VPR (Vertical Profile of Reflectivity, see section 4), this clearly results in a diminishing of the bright band effect, while increasing the lower rainfall at longer ranges.

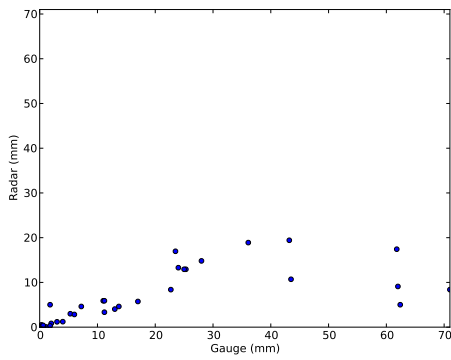
Comparing accumulations with rain gauges gives an indication off the performance of converting reflectivity to rainfall. The quality differs from precipitation type, and the current system is first and foremost optimised for stratiform summer precipitation. Figure 14-15 shows two examples of PPI(upper-left) and SRI(upper-right) from August and September where gauge data are also plotted in a combination with the accumulated precipitation products. The plots in figure 14 shows two scatter plots where the radar rainfall is plotted against gauge data, and the VPR method clearly gives an improvement of the estimated rainfall for this precipitation event.

The accumulations shows an improvement in the estimation of total rainfall, the scatter plots shows that low precipitation values are increased and the values are more linear. The gauge values without a corresponding radar pixel are left out of the plots and are probably missing due to radar coverage or overshooting.

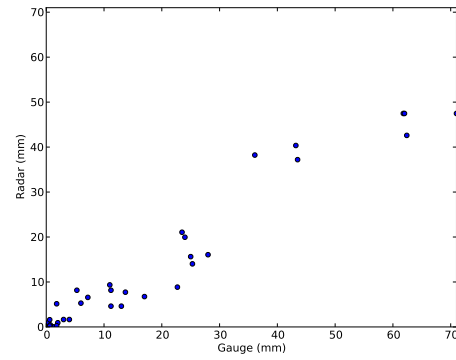
Figure 15 unveils some unwanted artifacts by combining VPR correction and clutter identification. The sea clutter in this product is correctly identified, and the clutter is not corrected by the VPR method, this is seen as the area with lower rainfall intensity in figure 15(b) at the south coast. Another artifact is the ship traffic that has not been detected by the quality control algorithms, this is seen in



(a) 24 hours PPI accumulation from the Rissa radar (b) VPR corrected accumulation, VPR reference height at 1.0 km



(c) $RMS = 21.2$



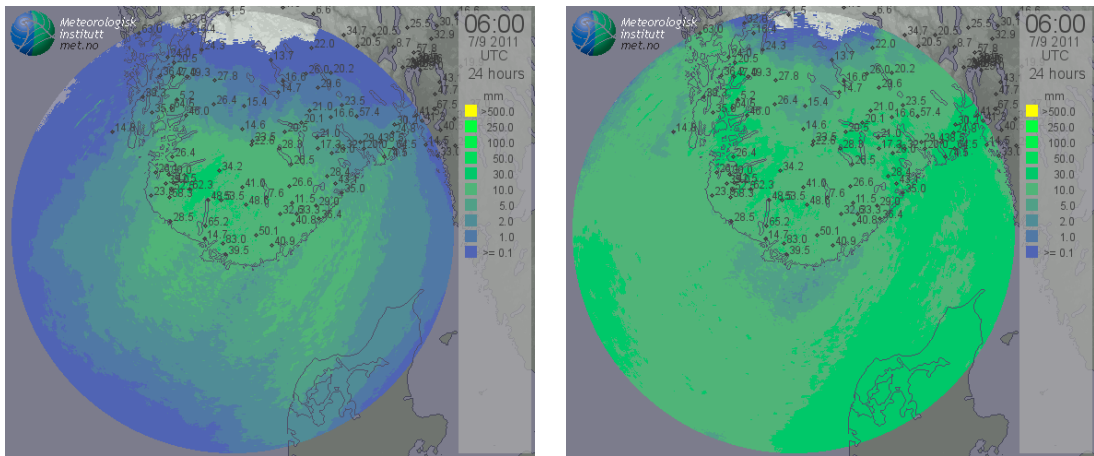
(d) $RMS = 5.0$

Figure 14: 24 hours radar accumulation from 17. August, plotted with gauge

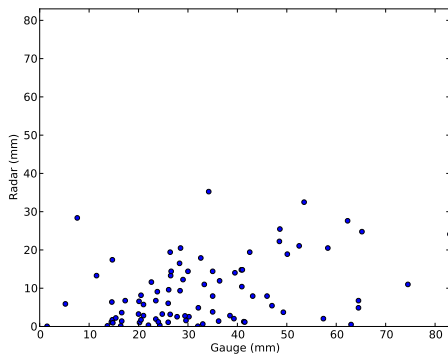
both figures as a strong straight line in the upper right of the scanning area, and the VPR method has increased this particular clutter.

Convective rainfall as in figure 16(a) is normally estimated quite well when correcting with a VPR method. The problem in this case is insufficient data near the radar, as the images show. This gives an incomplete vertical profile which has not been detected by the methods that control the quality of the profile before doing a correction. The scatter plots shows that the non-corrected (figure 16(c)) product is a reasonable estimate of the last 24 hour rainfall, while the corrected (figure 16(d)) product is clearly overestimating the precipitation.

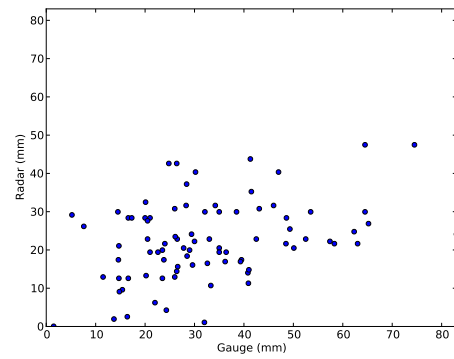
The rainfall data from gauges are downloaded from eKlima (<http://eklima.met.no>), and is considered to be accurate point measurements consisting of both manual and automatic stations. The manual 24 hour measurements are reported each day at 06:00 UTC.



(a) PPI accumulation showing reduced reflectivity with respect to distance from the radar (b) VPR corrected accumulation, shows a more uniform accumulation

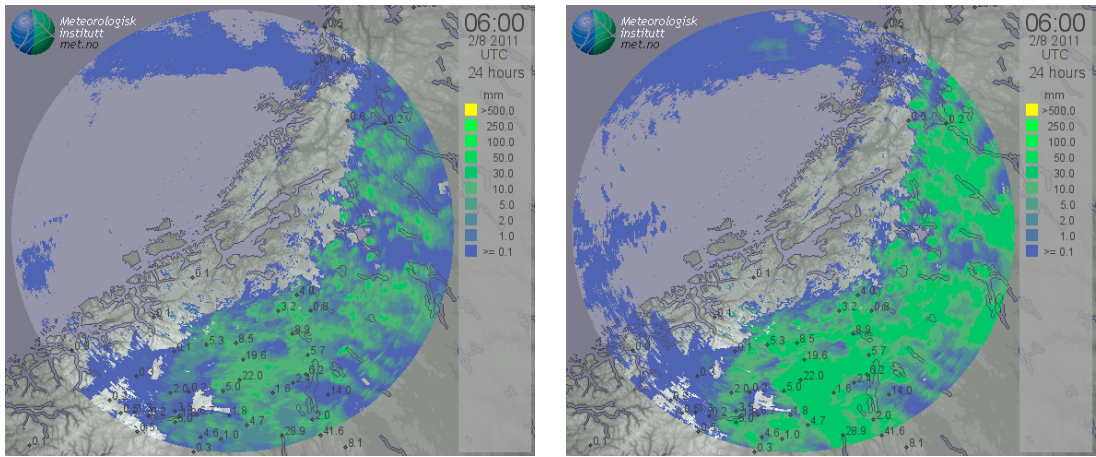


(c) RMS for PPI is equal to 28.0

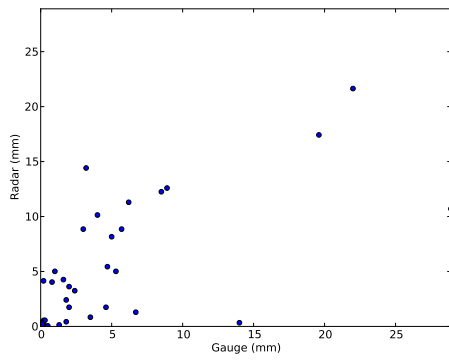


(d) Slightly improved scatter plot. RMS for SRI is equal to 18.5

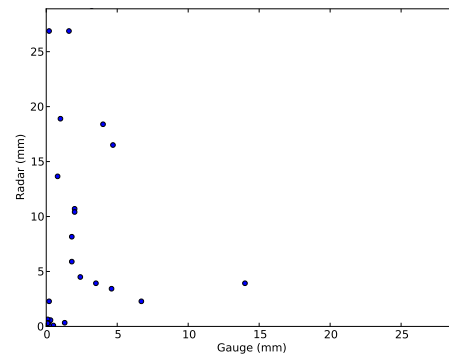
Figure 15: 24 hours radar accumulation from 7. September, plotted with gauge data.



(a) 24 hours PPI accumulation from Hegebostad radar (b) VPR corrected accumulation, VPR reference height at 1.0 km



(c) $RMS = 6.8$



(d) $RMS = 19.4$

Figure 16: Insufficient data near the radar gives an overestimation of the rainfall for this particular example.

4 Rain rate estimation

One of the main reason for errors in rainfall estimations from radar data is the vertical variability of the radar signal. The vertical variability is due to the different phase and type precipitation, this is clearly seen at the height of bright band where the precipitation changes phase from ice to droplets and resulting in an overestimation of the ground precipitation. Above the bright band the precipitation is dry, and therefore has a lower reflectivity.

The scan illustration in figure 9 on page 16 show how the beam changes height with respect to range, this means that the reflected signal near the radar can be drizzle or rain, while at longer ranges the reflected signal can be sleet, snow, graupel or hail. The scan strategy figure also shows that at long ranges from the radar, the center of the lowest beam is 5 km above ground level, and the measurement at this height will most likely be different from ground measurement.

There are several methods to overcome the difference between point measurements at ground level and measurements aloft, and the most promising method is to incorporate the vertical variability in the radar equation. This profile of vertical variability can be used to convert measurement aloft to a given reference level, thus making it independent of height and correct for the range-dependent error.

The authors of [1] gives a formulation on how to identify the so-called Vertical Profile of Reflectivity (VPR), which is an approach that takes the vertical variability into account. In short terms it assumes independent horizontal variability, such that

$$Z(\mathbf{x}, h) = Z(\mathbf{x}, h_0)z(h) \quad (3)$$

where $z(h)$ is the vertical profile of reflectivity that needs to be estimated and $Z(\mathbf{x}, h_0)$ is the estimated “ground truth” referred to by a reference level h_0 . The profile can be estimated by calculating the mean in the horizontal space, i.e

$$z(h) = \frac{\int_h Z(\mathbf{x}, h) d\mathbf{x}}{\int_{h_0} Z(\mathbf{x}, h_{h_0}) d\mathbf{x}}. \quad (4)$$

Once the estimated vertical profile of reflectivity is calculated, the correction factor

$$Z_{\text{corr}} = 1/z(h), \quad (5)$$

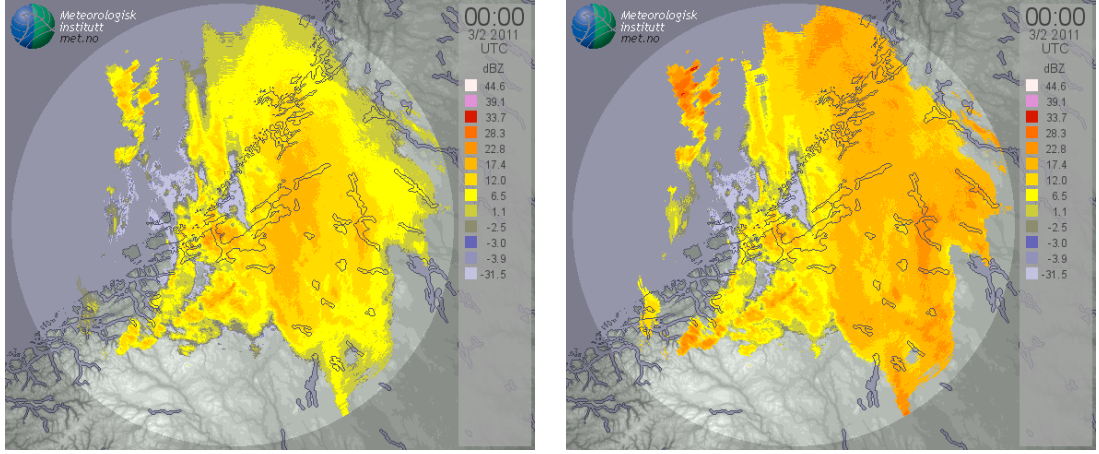
can be applied to $Z(\mathbf{x}, h)$ to achieve a VPR corrected product.

Clearly, the profiles need reflectivity data near the radar, preferably within 80 – 100 km in range according the scanning strategy in figure 9, this gives the profile generation sufficient data near the ground and aloft.

4.1 Examples

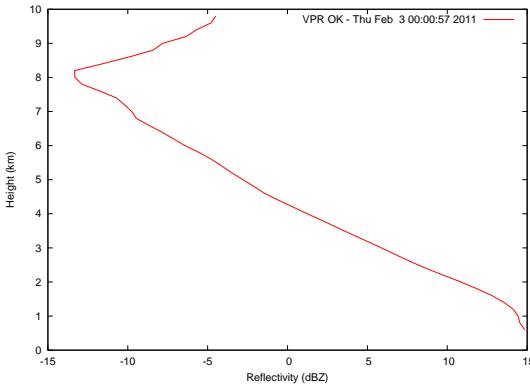
The following three examples (figure 17-19) show how the VPR method behave with different precipitation types. The first one gives an example on how the profile is decaying with height during a winter event, the precipitation is most likely snow since the profile has a negative gradient. The next example shows a strong convective event near the Oslo and Akershus area, and same with the previous example the profile is decaying this time, indicating that the precipitation is convective which is a typical situation in the summer. The last example shows a stratiform front with a bright band near 1.5 km in height.

The correction factor (dBZ-scale) illustrated in figure 17(d) and figure 18(d) where one clearly sees that the factor increases by range, hence correcting the distance-dependent error where lower measurement is increased up to a factor of 20 for snow and 10 for the convective case. For the stratiform precipitation case in figure 19, the bright band is the strong peak figure 19(c) located at 1.5 km in height, the correction factor shows how the correction is mitigating the signal within 60 – 70 km, hence correcting for the bright band, and increasing the signal further away from the radar.

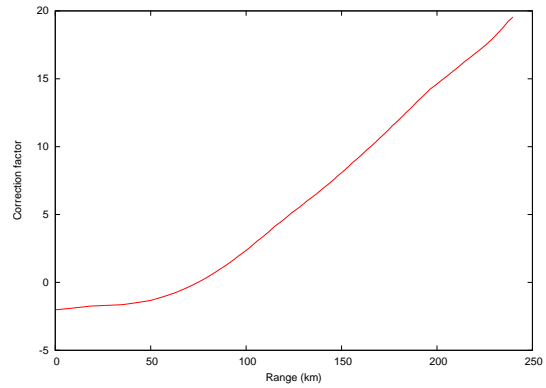


(a) 0.5° elevation at Rissa during the winter

(b) Corrected reflectivity by the vertical profile of reflectivity



(c) The vertical profile of reflectivity



(d) Z correction factor

Figure 17: Winter event from the Rissa radar, 3.2.2011-00:00.

The last example is a VPR corrected radar datasets accumulated into a 24 hour product, shown in figure 20 on page 28. Each radar product (PPI and SRI) has been converted to mm by the classical Z-R relationship

$$Z = AR^b \quad (6)$$

where A is equal to 200.0 and b is equal to 1.6. The figure also includes a plot of gauges within the radar cover area, showing the difference between gauges and the rain rate deduced by the ProRad software. The discontinuities in rate rate, in particular over the sea for this example, is due to the 15min time resolution and strong precipitation velocity.

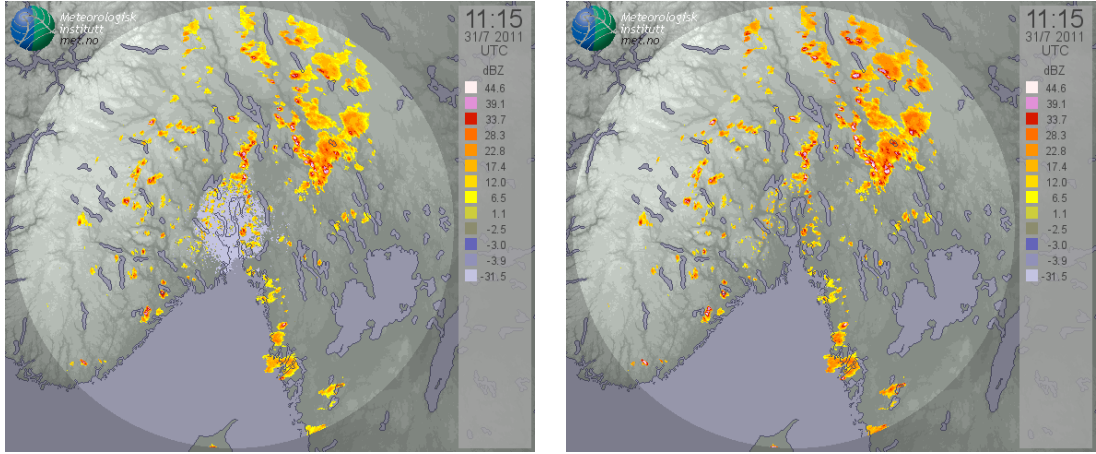
Tables 1 and 2 gives an error analysis of PPI and SRI products vs. rain gauge for 24 hour accumulation from both summer and winter precipitation. The errors are given as root mean square

$$RMS = \sqrt{\left(\frac{1}{n} \sum_{k=0}^n (RR_k - RG_k)^2\right)}, \quad (7)$$

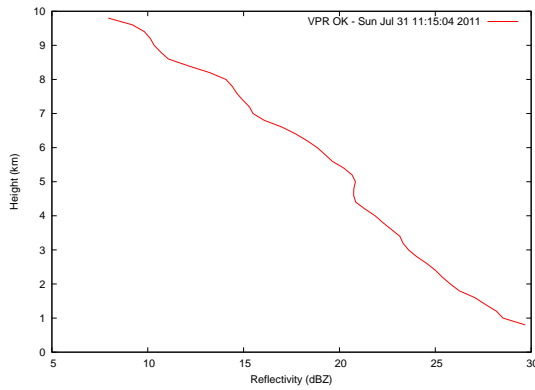
where the value k refers to the different simultaneous measurements between the rain rate pixel from a radar and a rain gauge. The correlation factor ρ

$$\rho_{RR, RG} = \frac{\text{cov}(RR, RG)}{\sigma_{RR} \sigma_{RG}}, \quad (8)$$

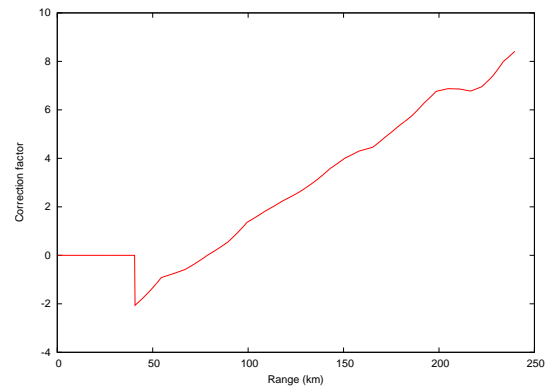
where $\text{cov}(RR, RG)$ is the covariance between RR and RG , and σ the respectively expected value. The $\rho_{RR, RG}$ factor varies from -1 to 1 , where the latter is perfect correlation and -1 the opposite.



(a) 0.5° elevation at Hurum during the summer, convective precipitation (b) Removed clutter and corrected the reflectivity with the VPR method



(c) The vertical profile of reflectivity



(d) Z correlation factor, the peak is due to insufficient data near the radar

Figure 18: Strong convective cells captured by the Hurum radar.

When ρ is approaching zero, the radar and gauge are more uncorrelated, this is often seen when the number of observations is small. The last error estimate, Mean Error (ME), gives an indication if the derived rain rate is over- or underestimating compared to the rain gauge. This is given as

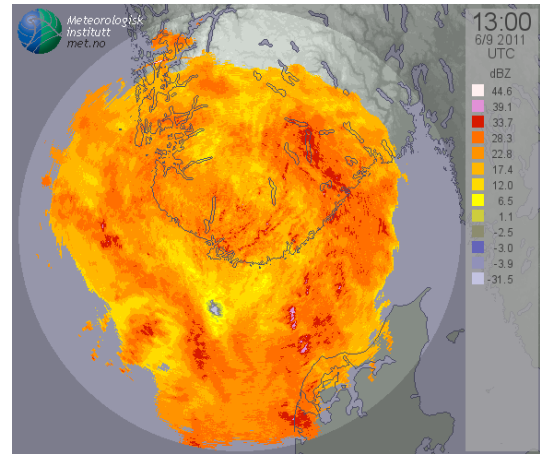
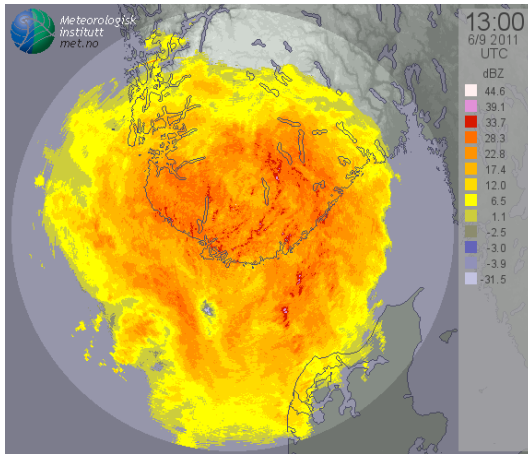
$$ME = \frac{1}{n} \sum_{k=0}^n (RG_k - RR_k). \quad (9)$$

Date	PPI RMSE	SRI RMSE	ρ_{PPI}	ρ_{SRI}	PPI ME	SRI ME	No. of obs
2011-06-02	21.7614	17.9617	0.5789	0.6761	-15.2153	-12.6487	31
2011-06-03							Missing data
2011-06-04							Missing data
2011-06-05							Missing data
2011-06-06	7.4341	7.2096	0.1820	0.2300	-2.5079	-1.9884	20
2011-06-07	15.4060	12.5429	0.4794	0.3689	-11.7208	-5.4548	81
2011-06-08	3.7329	2.7066	-0.1808	0.4604	-2.3201	-2.6145	4
2011-06-09	4.7090	7.2992	0.4875	0.1333	-3.0566	0.7754	76
2011-06-10	12.8595	9.0646	0.5791	0.7375	-7.5326	-3.2855	67
2011-06-11	5.4172	4.0647	0.6264	0.7339	-2.4526	-0.4160	20
2011-06-12	1.5614	1.2427	0.4430	0.6970	-0.1020	0.2397	24
2011-06-13	1.1616	1.9797	0.5754	0.1288	-0.4041	-0.8505	15
2011-06-14	4.3611	4.0046	0.5357	0.2406	-3.5573	-2.8265	60
2011-06-15	25.9501	19.9826	0.3272	0.5679	-15.8776	-10.4400	27
2011-06-16	8.3261	8.0122	-0.2591	-0.2815	-4.5958	-5.0431	9
2011-06-17	10.7822	11.9353	0.7768	0.4466	-8.0453	4.2630	77
2011-06-18	10.0589	7.1628	0.1004	0.3754	-7.6834	-2.6584	67
2011-06-19	6.3160	5.3954	0.0950	0.2200	-3.7970	-1.4047	40
2011-06-20	4.5165	5.9834	0.3254	0.3615	-2.1460	0.8424	60
2011-06-21	5.3887	5.6287	0.5044	0.7366	-2.2645	2.6327	25
2011-06-22							Missing data
2011-06-23	4.3273	7.5170	0.7686	0.7293	-1.8028	2.9193	51
2011-06-24	3.4178	18.6777	0.0609	-0.1210	-1.3807	9.4402	48
2011-06-25	7.1774	15.2042	0.3019	0.6955	-4.1587	9.1353	42
2011-06-26							Missing data
2011-06-27	11.2078	7.5785	0.6855	0.8438	-5.2443	-2.1060	74
2011-06-28	7.2444	5.2718	0.3701	0.6774	-3.2466	-0.1949	67
2011-06-29							Missing data
2011-06-30							Missing data

Table 1: RMS (radar rainfall vs. gauge) for summer precipitation from the Hegebostad radar. The radar accumulation is derived from 24 hour radar rainfall intensity.

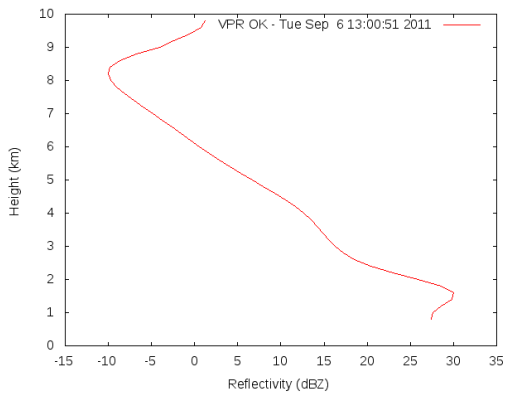
Date	PPI RMSE	SRI RMSE	ρ_{PPI}	ρ_{SRI}	PPI ME	SRI ME	No. of obs
2012-01-02	6.1673	5.8298	0.6149	0.6291	-4.3873	-4.3134	87
2012-01-03	2.4093	1.9013	0.7096	0.6904	-1.4681	-0.9597	21
2012-01-04	12.7897	11.6537	0.5426	0.4887	-11.3393	-10.3186	98
2012-01-05	4.0699	3.6601	0.1238	0.5634	-2.6013	-2.3909	25
2012-01-06							No data
2012-01-07	1.0797	1.0407	0.5378	0.5250	-0.7327	-0.7060	62
2012-01-08	2.0464	1.7893	0.5479	0.5040	-1.6413	-1.4963	56
2012-01-09	1.3738	1.2523	0.0961	0.1971	-0.9377	-0.7952	40
2012-01-10							No data
2012-01-11							No data
2012-01-12	5.1791	4.6620	0.4184	0.5134	-3.2616	-3.5905	41
2012-01-13	1.7654	1.6204	0.3612	0.4262	-0.8564	-0.8663	13
2012-01-14	1.5845	1.4440	-0.3606	0.0779	-1.0699	-0.6404	20
2012-01-15	0.3468	0.2691	-0.5071	0.0324	0.0700	0.1137	3
2012-01-16							Missing data
2012-01-17							Missing data
2012-01-18							Missing data
2012-01-19	3.3607	3.0342	0.1801	0.2565	-2.8536	-2.4832	91
2012-01-20							Missing data
2012-01-21	2.9680	2.6439	0.3987	0.2767	-2.2596	-2.0002	35
2012-01-22	12.4528	10.6382	0.7001	0.6387	-10.5221	-8.8406	69
2012-01-23	1.0013	0.9576	0.2542	0.2359	-0.5363	-0.3003	27
2012-01-24	2.3826	2.1612	-0.2093	0.0459	-1.0832	-0.9009	35
2012-01-25							Missing data
2012-01-26							Missing data
2012-01-27	4.9421	4.2461	-0.0962	0.1261	-2.8584	-2.4163	80
2012-01-28	6.3253	6.2310	-0.0962	0.0687	-3.2336	-3.1539	84
2012-01-29	3.4139	2.9224	0.0820	0.1319	0.9847	0.9612	27
2012-01-30	4.3615	3.6105	-0.2384	-0.1202	-2.0271	-2.1514	22
2012-01-31							Missing data

Table 2: RMS (radar rainfall vs. gauge) for winter precipitation from the Hurum radar. The radar accumulation is derived from 24 hour radar rainfall intensity.

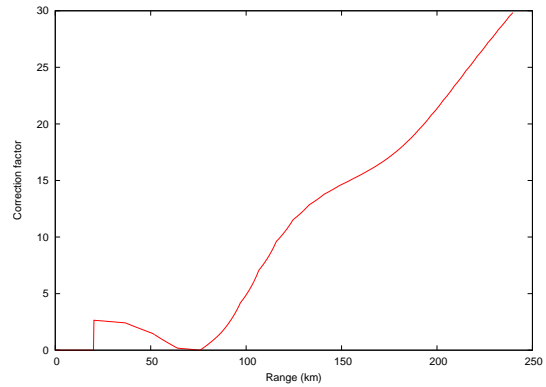


(a) 0.5° elevation at Hegebostad during the fall, stratiform precipitation

(b) Corrected the reflectivity with the VPR method

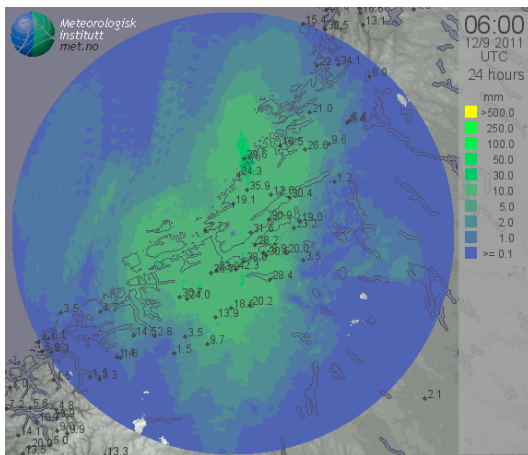


(c) The vertical profile of reflectivity

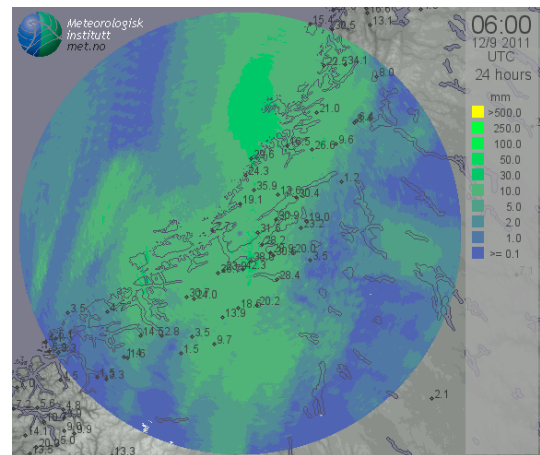


(d) Z correlation factor

Figure 19: Stratiform precipitation.



(a) Lowest elevation



(b) VPR corrected

Figure 20: 24 hour of accumulation from the Rissa radar.

A Radar propagation paths in ProRad

List of Symbols

s	Ground range
k	Section index number
r_k	Slant range
θ_k	Elevation angle
h_k	The height between to rays
ψ_r	Angle at the center of the earth
a_e	Effective earth radius

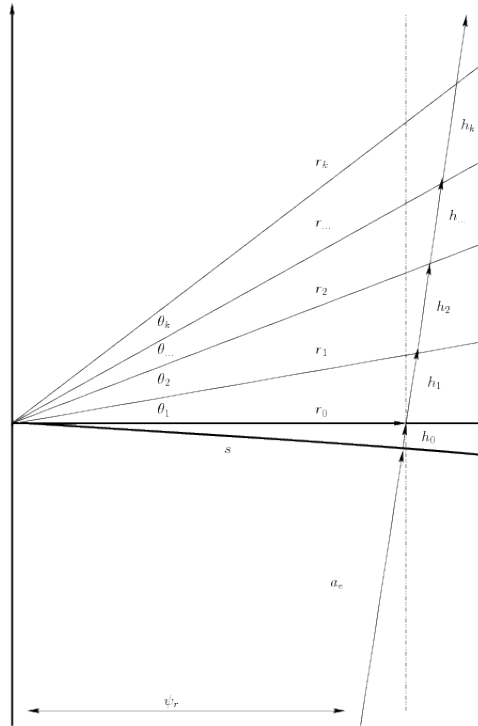


Figure 21: Radar paths and the variables describing the geometry

Figure 21 illustrates the radar geometry with earth curvature and how the variables relates. The angles θ_k where $k = 0, \dots, n$ is the elevation angles relative to the horizontally straight line. The straight lines h_k is the height which is slightly offset to the striped line, note that in the real world the angle ψ_r is very small compared to the figure. Note also that the skew lines is orthogonal to the earth curvature, such that the height above ground for a given point at ray k is the sum $\|h_k\| = \sum_{k=0}^n |h_k|$.

The height between the horizontally straight line ($\theta = 0$) and the ground range is found by solving

$$s^2 + (h_0 - a_e)^2 = a_e^2 \quad (10)$$

explicitly for h_0 assuming $h_0 \ll a_e$ such that

$$h_0 = \frac{s^2}{2a_e} \quad (11)$$

Two important equations are stated here from Doviak and Zrnić [4] that relate ground range s and height h to the known parameters that are stored in the meta structures; slant range r and elevation angle θ

$$h = a_e \left(\frac{\cos \theta}{\cos \theta + s/a_e} - 1 \right), \quad (12)$$

$$h = (r^2 + a_e^2 + 2ra_e \sin \theta)^{1/2} - a_e, \quad (13)$$

$$s = a_e \sin^{-1} \left(\frac{r \cos \theta}{a_e + h} \right). \quad (14)$$

A.1 Functions

The C prototype functions below will calculate different lengths depending on the input parameters.

Listing 11: prop_calc.h: Calculate slant range from height and ground range

```
double calc_r_from_h_s( const double h, const double s );
```

This function calculates the slant range from height h and ground range s .

Here r is calculated with the aid from the cosine law

$$r = \sqrt{r_0^2 + h_k^2 - 2r_0h_k \sin \xi_r} \quad (15)$$

where $r_0 = \sin \xi_r(h_0 + a_e)$ and $h_k = h - h_0$.

Listing 12: prop_calc.h: Calculate height from ground range and elevation angle

```
double calc_r_from_s_theta( const double s, const double theta );
```

This function calculates the slant range from ground range s and elevation angle θ .

r is solved from (14) where h is calculated by `calc_h_from_s_theta`.

Listing 13: prop_calc.h: Calculate slang range from height and elevation angle

```
double calc_r_from_h_theta( const double h, const double theta );
```

This function calculate the slant range from height h and elevation angle θ . It solves (13) explicitly for r , extracting the positive root,

$$r_+ = \frac{-2a_2 \sin \theta - \sqrt{(2a_e \sin \theta)^2 - 4(a_e^2 - (h + a_e)^2)}}{2} \quad (16)$$

Listing 14: prop_calc.h: Calculate elevation angle from height and ground range

```
double calc_theta_from_h_s( const double h, const double s );
```

This function calculate the elevation angle θ from height h and ground range s .

The calculation is based on finding the three lengths h_k, r_k, r_0 such that θ_k is achieved by the cosine law

$$\theta_k = \cos^{-1} \left(r_k^2 + r_0^2 - \frac{h_k^2}{2r_k r_0} \right), \quad (17)$$

r_0 is calculated by $\sin \xi_r(h_0 + a_e)$, $h_k = h - h_0$ and r by the `calc_r_from_h_s` function.

Listing 15: prop_calc.h: Calculate height from ground range and elevation angle

```
double calc_h_from_s_theta( const double s, const double theta );
```

This function calculate the height h from ground range s and elevation angle θ . Uses (12) directly.

A.2 Interpolation

A vertical cut $V \in \mathbf{R}^2$ is simply a hyperbolic section through the conic volume for a given azimuth degree. A value $Z(P) = Z(s, h) \in V$ in the figure 22 needs to be interpolated from surrounding values. The data are given by $Z(r, \theta)$ in the original volume is composed by $Z(r(s, h), \theta(s, h))$ where $r(s, h)$ and $\theta(s, h)$ are given functions in the Geometry section.

The grid (s_i, h_k) is discretized by e.g $s_i = s/ds$ and $h_k = h/dh$ for $i = 0, \dots, m$ and $k = 0, \dots, k$. Thus the vertical cut is created by looping through the discrete points with interpolated values.

The linear interpolation is calculated by a convex combination of 2 or more points, e.g for 2 points the distance is calculated for the 2 closest points in slant range $d_1 = \|r - r_k\|$ and $d_2 = \|r - r_{k+1}\|$. The weight factor $\omega = \frac{d_1}{d_1+d_2}$ gives the convex combination

$$Z(r, \theta) = \omega Z(r_k, \theta) + (1 - \omega) Z(r_{k+1}, \theta). \quad (18)$$

A.2.1 Functions

Listing 16: prop_calc.h: Interpolate Z data from nearest slant range

```
double interp_Z_rnearest( const td_radarprod_meta_st *meta, const double r,
    const size_t k );
```

This function gives the Z value for the nearest slant range r and elevation k (here k is an integer).

Listing 17: prop_calc.h: Interpolate Z data with linear distance combination

```
double interp_Z_rlinear( const td_radarprod_meta_st *meta, const double r,
    const size_t k );
```

This function gives the Z value with linear distance interpolation between the two closest slant ranges with (18).

Listing 18: prop_calc.h: Interpolate Z data with nearest point in space

```
double interp_Z_4nearest( const td_radarprod_meta_st *meta, const double s,
    const double theta );
```

This function will find the nearest point in the radar polar space within the given meta structure for the given parameters `ground_range` and `theta` (θ).

Listing 19: prop_calc.h: Interpolate Z value with bilinear combination

```
double interp_Z_bilinear( const td_radarprod_meta_st *meta, const double s,
    const double theta );
```

This function use a convex bilinear interpolation for the parameters `ground_range` and `theta` (θ) to calculate the point in the radar polar space.

Listing 20: prop_calc.h: Interpolate Z value with gaussian beam distribution

```
double interp_Z_gauss( const td_radarprod_meta_st *meta, const double
    r, const double theta );
```

Not yet implemented.

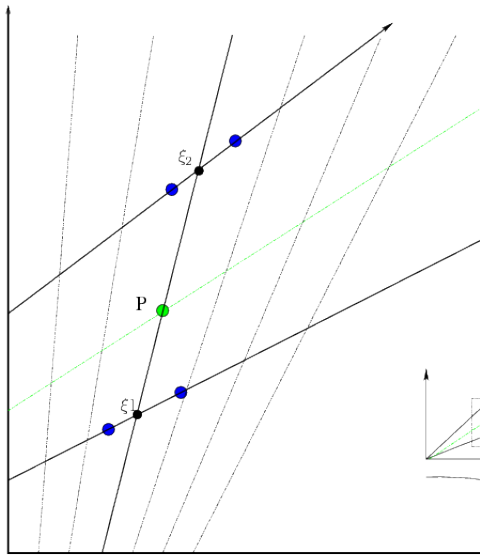


Figure 22: Bilinear interpolation. The green point is calculated by distance interpolation by the four blue points.

References

- [1] Hervé Andrieu and Jean Dominique Creutin. Identification of vertical profiles of radar reflectivity for hydrological applications using an inverse method. part 1: Formulation. *AMS*, 14(3):342–351, January 1994.
- [2] Alexis Berne, Guy Delrieu, Hervé Andrieu, and Jean-Dominique Creutin. Influence of the vertical profile of reflectivity on radar-estimated rain rates at short time steps. *Journal of Applied Meteorology*, 5(2):296–310, April 2004.
- [3] T. Bøvith, A. A. Nielsen, L. K. Hansen, R. S. Gill, and S. Overgaard. Detecting weather radar clutter by information fusion with satellite images and numerical weather prediction model output. In *Proceedings of the IEEE Geoscience and Remote Sensing Symposium (IGARSS) 2006*, volume I, pages 511–514, August 2006.
- [4] Richard J. Doviak and Dušan S. Zrnić. *Doppler Radar and Weather Observations*. Academic Press, 2 edition, 1993.
- [5] Mariá Franco, Rafael Sánchez-Diezma, and Daniel Sempere-Torres. Improving radar precipitation estimates by applying a vpr correction method based on separating precipitation types. In *5th European Conference on Radar in Meteorology and Hydrology*, Helsinki, Finland, 2008.
- [6] Rashphal S. Gill. Sea clutter removal using radar elevation dependent second order texture parameters. Scientific Report 07-01, DMI, Copenhagen, 2007.
- [7] Uta Gjertsen and Viel Ødegaard. The water phase of precipitation—a comparison between observed, estimated and predicted values. *Atmospheric Research*, 77(1–4):218–231, 2005.
- [8] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 1992.
- [9] Cathey Kessinger, Scott Ellis, and Joseph Van Andel. The radar echo classifier: a fuzzy logic algorithm for the wsr-88d. In AMS, editor, *3th Conference on Artificial Intelligence Applications to the Environmental Science*, volume 9 of 13. AMS, Februar 2003.
- [10] Jarmo Koistinen and Elena Saltikoff. Experience of customer products of accumulated snow, sleet and rain. In *Advanced weather radar systems*, pages 397–406. COST75, 1999.
- [11] Frank Silvio Marzano, Gianfranco Vulpiani, and Errico Picciotti. Rain field and reflectivity vertical profile reconstruction from c-band radar volumetric data. *Geoscience and Remote Sensing*, 42(5):1033–1046, May 2004.
- [12] Martin S. Grønsleth and Christoffer A. Elo. Assimilating radar observations into the next generation numerical weather prediction model. *NOTUR II*, META Number 4:17–19, 2011.
- [13] Martin S. Grønsleth and Roger Randriamampianina. Assimilation of radar reflectivity data in harmonie. met.no report no. 1/2012, The Norwegian Meteorological Institute, 2012.
- [14] G. S. Pankiewicz, C. J. Johnson, and D. L. Harrison. Improving radar observations of precipitation with a meteosat neural network classifier. *Meteorology and Atmospheric Physics*, 76(1-2):9–22, 2001.
- [15] M.A. Rico-Ramirez and I.D. Cluckie. Bright-band detection from radar vertical reflectivity profiles. *International Journal of Remote Sensing*, 28(18):4013–4025, September 2007.

- [16] Rafael Sánchez-Diezma, Isztar Zawadzki, and Daniel Sempere-Torres. Identification of the bright band through the analysis of volumetric radar data. *Journal of Geophysical Research*, 105(D2):2225–2236, 2000.
- [17] J. A. Sethian. A fast marching level set method for monotonically advancing fronts. In *Proceedings of the National Academy of Sciences of the United States of America*, volume 93, pages 1591–1595, 1996.
- [18] Bertrand Vignal and Hervé. Identification of vertical profiles of reflectivity from volume scan radar data. *Journal of Applied Meteorology*, 38(8):1214–1228, August 1999.
- [19] Bertrand Vignal and Witold F. Krajewski. Large-sample evaluation of two methods to correct range-dependent error for wsr-88d rainfall estimates. *Journal of Hydrometeorology*, 2(5):490–504, October 2001.