Uncertainty propagation in stereo matching using copulas

Roman Malinowski, Sébastien Destercke, Loïc Dumas, Emmanuel Dubois, Emmanuelle Sarrazin

Uncertainty Propagation in Stereo Matching using Copulas

Roman Malinowski[a,b,c], Sébastien Destercke[c], Loïc Dumas[b], Emmanuel Dubois[a], Emmanuelle Sarrazin[a]

[a] *Centre National D'Etudes Spatiales (CNES), 18 Avenue Edouard Belin, Toulouse, 31000, France*
[b] *CS, 6 rue Brindejonc des Moulinais, Toulouse, 31000, France*
[c] *Université de technologie de Compiègne, Heudyasic, 57 avenue de Landshut, Compiègne, 60203, France*

Abstract

This contribution presents a concrete example of uncertainty propagation in a stereo matching pipeline. It considers the problem of matching pixels between pairs of images whose radiometry is uncertain and modeled by possibility distributions. Copulas serve as dependency models between variables and are used to propagate the imprecise models. The propagation steps are detailed in the simple case of the Sum of Absolute Difference cost function for didactic purposes. The method results in an imprecise matching cost curve. To reduce computation time, a sufficient condition for conserving possibility distributions after the propagation is also presented. Finally, results are compared with Monte Carlo simulations, indicating that the method produces envelopes capable of correctly estimating the matching cost.

Keywords: Possibility distributions, Copulas, Stereo Matching

1. Introduction

This contribution presents a concrete example of uncertainty propagation in the context of photogrammetry, and more specifically in the crucial step of

Email addresses: roman.malinowski@hds.utc.fr (Roman Malinowski), sebastien.destercke@hds.utc.fr (Sébastien Destercke), loic.dumas@csgroup.eu (Loïc Dumas), emmanuel.dubois@cnes.fr (Emmanuel Dubois), emmanuelle.sarrazin@cnes.fr (Emmanuelle Sarrazin)

matching cost computation [1]. Recent research in this field aims to estimate the uncertainty associated with dense stereo matching in specific steps of the pipeline [2, 3, 4], or with end-to-end methods [5]. Those methods either estimate the uncertainty *a posteriori* without considering the uncertainty of the input data, or are not explainable, as for the case of deep-learning based methods. In [6], authors have estimated the uncertainty using a precise density function, and adapt their matching strategy to minimize this uncertainty. Instead, we argue in favor of using imprecise models to represent the uncertainty regarding our data, due to the noise, partial volume effect and various processing steps that can affect the images. In this contribution, we will use belief functions, and more specifically possibility distributions [7], to model the uncertainty on image intensities. We use copulas to characterize the dependency between models of uncertainty, and to create multivariate uncertainty models. The uncertainty is then propagated using a cost function from classical stereo matching problems. The resulting belief function [8, 9] allows us multiple envelopes with different degrees of plausibility to be defined, centered on the matching cost computed without uncertainty. Those envelopes are validated using Monte Carlo simulations with multiple models of noise on the input images.

Section 2 presents the stereo matching problem and the considered sources of uncertainty. Section 3 describes the univariate uncertainty models considered in this study, notably those used on the input images. Section 4 then deals with the construction of a joint model definition through dependency models that are copulas. Section 5 explains the method used to propagate the imprecise models based on copulas and details the specific case of propagation through a matching cost function. A sufficient condition for conserving possibility distributions through the propagation is also described. Finally, we present the resulting belief functions as well as Monte Carlo simulations for different copulas in Section 6. Note that this paper extends a previously published paper [10], notably by discussing in more details the relation between copula and possibility distributions, and by providing complete experiments on large-scale images, thereby demonstrating the applicability of the developed models and propagation tools (that are also more detailed) to actual stereo matching issues.

2. The Stereo Matching Problem

2.1. Dense Matching

Stereo Matching is one of the main steps when reconstructing 3D models from pairs of images by photogrammetry. For a complete description and review of stereo methods, we refer to [1]. In photogrammetry, images of the same scene are acquired from different points of view, and depth information of each pixel is retrieved by evaluating its displacement between images. Figure 1 presents a pair of images and highlights the position of the same pixel in both images. This displacement is called "disparity". We consider the case with two images, referred to as the *left* and *right* images. Images are often rectified so that the displacement of pixels can only occur in one direction, usually horizontally [11]. This can be a source of uncertainty on pixel values. The rectification allows us to restrict the search for a pixel's match to a single row. The problem could be briefly summarised as follows: given a pair of left-right images (I_L, I_R), determine for every pixel $p_L(i,j) \in I_L$ of the left image the disparity d of position, allowing its homologous pixel $p_R(i, j-d) \in I_R$ in the right image to be found. By knowing the disparity d of an object, the displacement B and focal length f of the camera, the depth z of the object is computed using the following relation:

$$z = \frac{Bf}{d}\,. \tag{1}$$

In practice, not all pixels of the left image have a corresponding pixel in the right image. Some zones can become occluded by an object when changing the camera's point of view, or similarly, pixels that where hidden behind an object in the left image might appear in the right image. Those occluded zones can be identified *a posteriori* [12], so we will not consider this issue here.

2.2. Cost Functions

For each pixel, we evaluate whether pixels of the other image are good candidates for a match by measuring their similarity. Cost functions measure the similarity between two patches centered on the pixels that are compared. Patches are usually squared windows, although other shapes can be considered [13]. Evaluating the matching cost of a function between a patch in the left image and a list of potential matches in the right image, gives a *cost curve*. In the case where the cost function measures a distance, low values represent

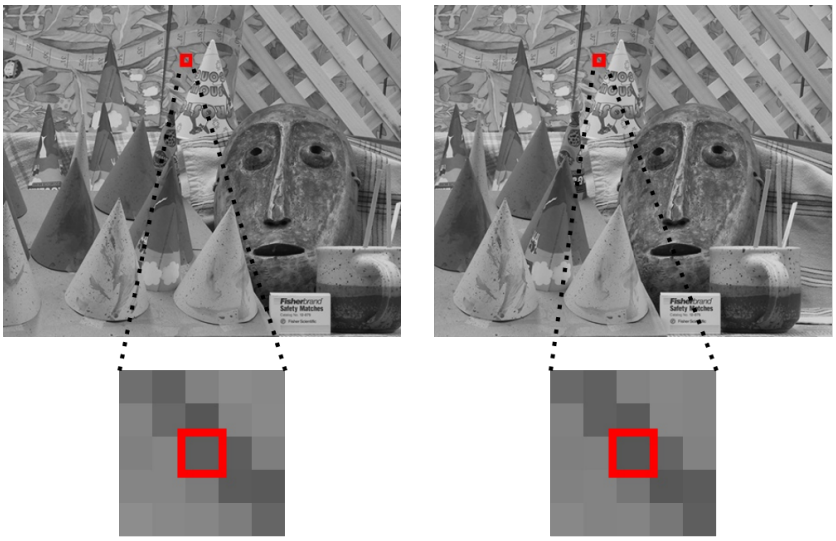


Figure 1: Homologous pixels in a pair of images

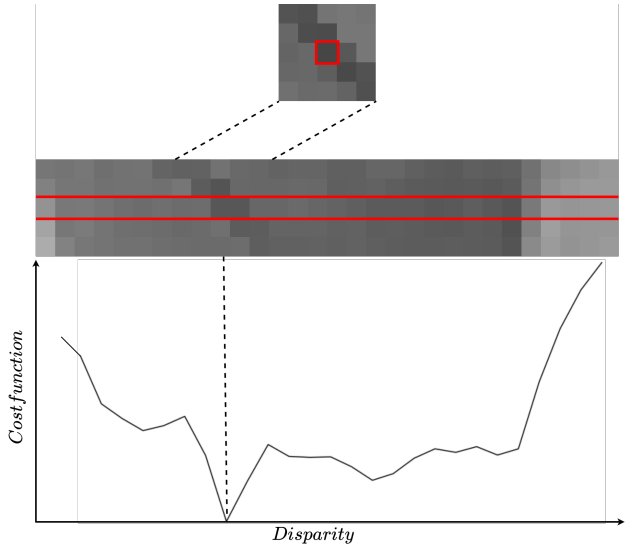


Figure 2: Patch comparison to find the disparity between two images. From top to bottom: left image patch, patches from the right image, cost curve

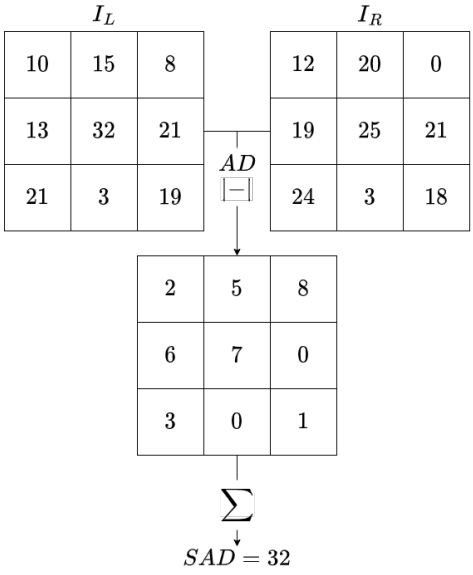


Figure 3: Example of the SAD cost function

a strong similarity, and the correct disparity is determined by finding the minimum of a cost curve. In the case where the cost function measures a correlation, the correct disparity is determined by finding the maximum of the cost curve. In this article, we consider a cost function measuring a distance, and look for its minimum. An example of this procedure is shown in Figure 2. The top and middle figures present a patch of the left image and a row of the right image, where a potential match is sought. The bottom plot shows the corresponding cost curve, where each patch of the right image row is compared with the left image patch. The minimum of the cost curve indicates the correct disparity.

In this article, we will focus on using a basic Sum of Absolute Differences (SAD) cost function, defined as follows. Given patches $W_L \subset I_L$ and $W_R \subset I_R$ of the same shape with n pixels (usually squares):

$$\mathrm{SAD}(W_L, W_R) = \sum_{(p_i, q_i) \in (W_L, W_R)} |I_L(p_i) - I_R(q_i)|\,, \tag{2}$$

where p_i and q_i are pixels at the same position i in their patch. For convenience purposes, we will refer to the Absolute Difference as AD. This cost function

is not ideal, but it is preferred here for its didactic properties. An illustration of the SAD cost function can be found in Figure 3.

An ideal cost function would generate a cost curve with a unique minimum corresponding to the correct disparity. In practice, such a function is hard to determine. There is no guarantee that the minimum is unique, nor that it corresponds to the correct disparity. Different cost functions have been proposed to better identify the correct match [14], some being robust to small variations of intensities [15], or determined using advanced method such as deep learning approaches [16]. We have chosen the SAD method to focus on simplicity and to ease didactic explanations regarding uncertainty propagation. Other cost functions can produce better results but with less facility to explain this paper's methodology.

2.3. Sources and Modeling of Uncertainty in Stereo Matching

In Section 2.1, we stated that images can be rectified to ensure horizontal displacement of pixels. It is often the case in remote-sensing, where images are taken by airplane or satellites [17]. This pre-processing step, added to the noise of the sensor taking the image, generates uncertainty surrounding the value of every pixel. Our aim is to propagate this uncertainty through the matching cost evaluation, in order to have an estimation of the uncertainty attached to the matching cost of two patches. Knowing this uncertainty can help in better identifying the correct disparity [6], for instance in the case where multiple minima of the cost curve exist, a good strategy may be to select the disparity with the least uncertainty. The following section presents models of uncertainty used in this contribution.

3. Uncertainty Models

The noise of the sensor and the uncertainty due to the pre-processing steps make it difficult to specify a precise probability model. Consequently, an *imprecise* model is preferred for this problem. Definitions regarding the imprecise probability framework are now presented.

3.1. Belief Functions

Consider a random variable X defined over a measurable space $\mathcal{X}$. A probability mass function m over a space $\mathcal{X}$ is a mapping $\mathcal{P}(X) \rightarrow [0, 1]$ satisfying:

$$m(\emptyset) = 0, \quad \sum_{X \subseteq \mathcal{X}} m(X) = 1\,, \tag{3}$$

where $\mathcal{P}(\mathcal{X})$ is the power set of $\mathcal{X}$. The subsets of $\mathcal{X}$ whose mass is strictly positive are referred with the letter a. For clarity, the exponent of focal sets will refer to the space or variable they are defined over, and the subscripts will refer to an order (if it exists). For instance, a_3^X refers to the third focal set of variable X. From a mass function, it is possible to define a belief Bel and a plausibility Pl function, which are mappings $\mathcal{P}(X) \to [0,1]$ defined as:

$$\begin{aligned} \forall A \subseteq \mathcal{X}, \mathrm{Bel}(A) &= \sum_{a \subseteq A} m(a)\,, \\ \forall A \subseteq \mathcal{X}, \mathrm{Pl}(A) &= \sum_{a, a \cap A \neq \emptyset} m(a)\,, \end{aligned} \tag{4}$$

Similarly, a belief function can also be interpreted as the lower bound of a convex set of probability measures $\mathcal{M}$, called a *credal set*, defined as:

$$\mathcal{M} = \{P \mid P(A) \geqslant \mathrm{Bel}(A), \forall A \subseteq \mathcal{X}\}\,. \tag{5}$$

3.2. Possibility Distributions on Image Intensities

We work with grayscale images with a quantification of intensity levels in $[0, 255]$, corresponding to our measurable space $\mathcal{X}$. In this paper, we make the hypothesis that the value of a pixel cannot be more than 1 intensity level away from its observed value, and that the observed value is the most plausible. This is due to the quantification of the observed radiometry into integers, but it could be extended to also take into consideration the uncertainty resulting from the rectification step of section 2.1. We keep a simple hypothesis for didactic purposes. We thus chose to model the uncertainty of the intensity of every pixel $p \in I_L, I_R$ by a possibility distribution π centered on the observed intensity i_p:

$$\pi(i_p) = 1, \quad \pi(i_p \pm 1) = \alpha\,, \tag{6}$$

with $\alpha \in [0,1]$. $\pm$ means that we consider both the plus or minus values. In our simulation, $\alpha = 0.3$ for pixels in the left image, and $\alpha = 0.4$ for pixels in the right image. We use different values of *alpha* for the left and right image because the uncertainty model could change from one image to the other, due to a different exposure, different noise or different calibration of our camera. This model is equivalent to state that we accept every probability with support in $[i_p - 1, i_p + 1]$ and whose probability measure P verifies $\{P(A) \leqslant \sup_{i \in A} \pi(i)\}$ as an acceptable model for our uncertainty. It has been shown [7] that to every possibility distribution corresponds a minitive belief

function (also called a necessity function) whose mass distribution function for every focal set a^p is in our case:

$$\begin{aligned} m_p(a_1^p = [\![i_p, i_p]\!]) &= 1 - \alpha\,, \\ m_p(a_2^p = [\![i_p - 1, i_p + 1]\!]) &= \alpha\,, \end{aligned} \tag{7}$$

with $[\![\cdot, \cdot]\!]$ referring to integer intervals. In particular, $[\![i_p, i_p]\!]$ correspond to the singleton $\{i_p\}$

It is noteworthy that in this problem of disparity estimation, we only consider the uncertainty of our input image intensities, but do not consider the uncertainty regarding our cost function's ability to correctly identify the true disparity as its minimum. In other words, we don't take into consideration the uncertainty resulting from the difference between "two patches are very similar" and "the pixels at the center of the patches are homologous". To better illustrate this, consider a case where two pixels should be matched, but the pixels in the patches surrounding them are dissimilar. Then the cost function between those two patches would be high, and there can be another patch with a lower cost function that would be wrongly selected as it is the minimum of the cost curve.

4. Joint Uncertainty Models

4.1. Copulas as Dependency Models

When propagating probability densities, one has to take into account the dependency between the different sources of uncertainty. Copulas are great tools to model the dependency between variables, as it has been shown that they can represent any kind of probabilistic dependency [18]. A copula, or N-copula, is a mapping $C : [0, 1]^N \to [0, 1]$ satisfying a number of properties [19] expressed here in the N dimensional case. For all k in $[\![1, N]\!]$ and for all $(u_1, \ldots, u_N)$ in $[0, 1]^N$:

$$C(u_1, \ldots, u_{k-1}, 0, u_{k+1}, \ldots, u_N) = 0\,, \tag{8}$$

$$C(1, \ldots, 1, u_k, 1, \ldots, 1) = u_k\,. \tag{9}$$

It also needs to be N-increasing, meaning that for all $U = (u_1, \ldots, u_N)$ in $[0, 1]^N$, and for all $V = (v_1, \ldots, v_N)$ in $[0, 1]^N$, such that $\forall k \in [\![1, N]\!], u_k \leqslant v_k$:

$$\sum_{\substack{(w_1,\ldots,w_N)\in \\ \otimes_k\{u_k,v_k\}}} (-1)^{|\{k \;|\; w_k=u_k\}|} C(w_1, \ldots, w_k) \geqslant 0\,, \tag{10}$$

with $\otimes$ representing the Cartesian product of sets, and $|\cdot|$ corresponding to the cardinal of a set. The left term of the previous inequality is also called the hyper-volume or H-volume of the copula. The inequality then reads as the H-volume of the copula of every segment of the unit hypercube is positive.

A copula can also be interpreted as a multivariate distribution function whose marginals are uniform on the unit interval.

Some famous copulas include:

- The product copula representing independence:
  $C_\Pi = \Pi_k u_k$
- The upper Fréchet-Hoeffding bound representing complete co-monotonicity:
  $C_{\min} = \min_k \ u_k$
- the Gaussian copula with correlation matrix R:
  $C_R = \Phi_R(\Phi^{-1}(u_1), \ldots, \Phi^{-1}(u_N))$, where Φ_R is the joint multivariate distribution function of a Gaussian variable with correlation matrix R, and Φ^{-1} is the inverse distribution function of a univariate Gaussian variable.

Sklar's theorem states that every multivariate cumulative distribution function (CDF) $G : \otimes_k \mathcal{X}_k \to [0,1]$ can be expressed by means of its marginal CDFs $F_k : \mathcal{X}_k \to [0,1], k \in [\![1, N]\!]$ and a copula C:

$$G = C(F_1, \ldots, F_N)\,. \tag{11}$$

The reverse implication is also true, meaning that joining any number of univariate CDFs with a copula returns a correctly defined multivariate CDF.

4.2. Building Joint Belief Functions with Copulas

In the following paragraph, we will explain how we use copulas to join the uncertainty models. Let us first explain it in the precise case. Let $\mathcal{X} = x_1, \ldots, x_{N^X}$, $\mathcal{Y} = y_1, \ldots, y_{N^Y}$ and $\mathcal{Z}$ be three discrete spaces, and let X, Y be two discrete random sets taking values in $\mathcal{X}, \mathcal{Y}$ respectively, with respective CDFs $F_X : \mathcal{X} \to [0,1], F_Y : \mathcal{Y} \to [0,1]$ and whose dependency can be represented by a copula C. To avoid heavy notations, we will refer to $H^{u_2,v_2}_{u_1,v_1}$ as the H-volume of the copula C over $[u_1, u_2] \otimes [v_1, v_2] \subseteq [0,1]^2$. Then we know that the joint probability mass distribution $p : \mathcal{X} \otimes \mathcal{Y} \to [0,1]$ is

defined as

$$\forall (i,j) \in [\![1, N^X]\!] \otimes [\![1, N^Y]\!],$$
$$p(x_i, y_j) = H^{F_X(x_i), F_Y(y_j)}_{F_X(x_{i-1}), F_Y(y_{j-1})}, \tag{12}$$

with the convention that $F_X(x_0) = F_Y(y_0) = 0$.

Let us now consider the case where we do not know the CDFs of X and Y, but only the belief functions $\mathrm{Bel}_X, \mathrm{Bel}_Y$ and their associated mass functions $m_X : \mathcal{P}(\mathcal{X}) \to [0,1], m_Y : \mathcal{P}(\mathcal{Y}) \to [0,1]$. Joining belief functions can be done in a similar manner as in (12), replacing the CDFs by cumulated masses [20]. In order to do so, we need to fix an arbitrary order over the focal sets of m_X and m_Y. Let us suppose we have such an order and that the focal sets of m_X are $\{a^X_1, \ldots, a^X_{N^X_a}\}$ and those of m_Y are $\{a^Y_1, \ldots, a^Y_{N^Y_a}\}$. Then the joint mass for X and Y is defined as:

$$m_{XY}(a^X_i, a^Y_j) = H^{\sum_{k=1}^{i} m_X(a^X_k), \sum_{k=1}^{j} m_Y(a^Y_k)}_{\sum_{k=1}^{i-1} m_X(a^X_k), \sum_{k=1}^{j-1} m_Y(a^Y_k)}. \tag{13}$$

We refer to a previous publication for details about this method and details on other ways of aggregating credal sets with a copula, as well as the importance of the order over focal sets [21]. Detailed work on joining random sets and with copulas can be found in [8, 9].

An important thing to note here is that a copula does not bare the same meaning when used with precise models in (12) and when used with imprecise models as in (13). In the precise case, the copula will encode the dependency between the values of the random variables. For instance in the case of comonotonicity inside an image patch, using the minimum copula $C_{\min}$ would mean that a high value of a pixel is strongly correlated to high intensities of neighbouring pixels. In the imprecise case and according to our definition, the copula will encode the dependency between degrees of belief regarding the values of random sets (supposing that there exist such underlying random sets behind our models). Using the minimum copula $C_{\min}$ in that case indicates that a high belief regarding a pixel's intensity is correlated to a high belief regarding its neighbours. However, the confident values of pixels could be very low and very high, which would not be the case using equation (12).

4.3. On copulas, possibility distributions and inclusion

Properties regarding copulas can be sometimes difficult to grasp, even more when they are used to join imprecise models such as possibilities. We

explore here some observations to better grasp the behaviour of copulas in an imprecise setting. We aim to be as didactic as possible, in order to avoid any misconceptions regarding the use of possibilities and copulas.

We look in particular to relations of inclusions/dominance one could think hold in general. This seems quite natural, given the nested nature of the focal sets of possibility distributions. We however show in this section that many such intuitions are actually invalid.

4.3.1. Domination relation on copulas does not propagate to possiblities

Contrary to what one could think, a domination relation on copulas does not create a domination relation on joint possibilities. Statement such as "$C' \leq C$ then $\pi' \leq \pi$" cannot be made. Let us illustrate this with an example, by taking two symmetrical uni-modal possibility distributions, and propagating them with a copula C and a mapping f. Let $\overline{x}, \overline{y}, \delta_x, \delta_y$ be reals such that $\delta_x > 0$ and $\delta_y > 0$. Let X, Y be two random sets with mass functions m_X, m_Y, noted:

$$\begin{array}{rclcrcl} m_X([\overline{x},\overline{x}]) & = & m_1^X, & & m_Y([\overline{y},\overline{y}]) & = & m_1^Y, \\ m_X([\overline{x}-\delta_x,\overline{x}+\delta_x]) & = & m_2^X, & & m_Y([\overline{y}-\delta_y,\overline{y}+\delta_y]) & = & m_2^Y\,. \end{array}$$

These mass functions induce two symmetric uni-modal possibility distributions, which can be propagated into another possibility distribution using a copula C and a linear combination $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ (see Section 5.3 for details on why the linear combination yields a possibility distribution). The joint mass m_{XY} can be computed using C and the natural order on focal sets of m_X and m_Y:

$$\begin{array}{rcl} m_{XY}^C([\overline{x},\overline{x}]\times[\overline{y},\overline{y}]) & = & C(m_1^X, m_1^Y)\,, \\ m_{XY}^C([\overline{x}-\delta_x,\overline{x}+\delta_x]\times[\overline{y},\overline{y}]) & = & m_1^Y - C(m_1^X, m_1^Y)\,, \\ m_{XY}^C([\overline{x},\overline{x}]\times[\overline{y}-\delta_y,\overline{y}+\delta_y]) & = & m_1^X - C(m_1^X, m_1^Y)\,, \\ m_{XY}^C([\overline{x}-\delta_x,\overline{x}+\delta_x]\times[\overline{y}-\delta_y,\overline{y}+\delta_y]) & = & 1 - m_1^X - m_1^Y + C(m_1^X, m_1^Y)\,. \end{array}$$

Computing the corresponding possibility $\pi_{f(X,Y)}^C$ is achieved by looking at the Plausibility function

$$\pi_{f(X,Y)}^C(f(x,y)) = \mathrm{Pl}(f(x,y)) = \sum_{A \;|\; (x,y)\cap A \neq \emptyset} m_{XY}^C(A)\,.$$

Consider events $[\overline{x}-\delta_x, \overline{x}+\delta_x] \times [\overline{y}, \overline{y}]$ and $[\overline{x}, \overline{x}] \times [\overline{y}-\delta_y, \overline{y}+\delta_y]$. For simplicity, let say that $a, b, c, \delta_x, \delta_y$ are all positive and:

$$f(x, y) = ax + by + c$$

Then,

$$\begin{aligned} f([\overline{x}-\delta_x, \overline{x}+\delta_x] \times [\overline{y}, \overline{y}]) &= [a\overline{x} + b\overline{y} + c - a\delta_x, a\overline{x} + b\overline{y} + c + a\delta_x] \\ f([\overline{x}, \overline{x}] \times [\overline{y}-\delta_y, \overline{y}+\delta_y]) &= [a\overline{x} + b\overline{y} + c - b\delta_y, a\overline{x} + b\overline{y} + c + b\delta_y] \end{aligned}$$

The results are two intervals with the same center $a\overline{x}+b\overline{y}+c$, and with sizes $2\delta_x$ and $2\delta_y$ respectively. There necessarily is an inclusion relation between them, and we will only deal with the first one as the second inclusion is similar. The same results hold for any real value of a, b and c.

If $f([\overline{x}-\delta_x, \overline{x}+\delta_x] \times [\overline{y}, \overline{y}]) \subseteq f([\overline{x}, \overline{x}] \times [\overline{y}-\delta_y, \overline{y}+\delta_y])$, then in particular:

$$\begin{aligned} \pi^C_{f(X,Y)}(f(\overline{x} \pm \delta_x, \overline{y})) &= 1 - C(m_1^X, m_1^Y) \\ \pi^C_{f(X,Y)}(f(\overline{x} \pm \delta_x, \overline{y} \pm \delta_y)) &= 1 - m_1^X - m_1^Y + C(m_1^X, m_1^Y)\,. \end{aligned}$$

The case where the inclusion is reversed is similar. Evaluating those values for two copulas C, C' such that $C < C'$, we remark that:

$$\begin{aligned} \pi^C_{f(X,Y)}(f(\overline{x} \pm \delta_x, \overline{y})) &> \pi^{C'}_{f(X,Y)}(f(\overline{x} \pm \delta_x, \overline{y}))\,, \\ \pi^C_{f(X,Y)}(f(\overline{x} \pm \delta_x, \overline{y} \pm \delta_y)) &< \pi^{C'}_{f(X,Y)}(f(\overline{x} \pm \delta_x, \overline{y} \pm \delta_y))\,, \end{aligned}$$

simply because the possibility degree $\pi^C_{f(X,Y)}(f(\overline{x} \pm \delta_x, \overline{y}))$ is decreasing with $C(m_1^X, m_1^Y)$, while it is increasing in the same variable for for $\pi^C_{f(X,Y)}(f(\overline{x} \pm \delta_x, \overline{y} \pm \delta_y))$, meaning that domination properties on the copulas are not transferred to the possibilities.

4.3.2. Applying copulas to mass functions against point-wise application

Equation 5 reminds us that a mass function can also be interpreted as a set of probability $\mathcal{M}$. When considering possibility distributions π_1 and π_2 and their associated probability sets $\mathcal{M}_1$ and $\mathcal{M}_2$, one is tempted to think that applying a copula as in Equation 13 could have some inclusion relationship with a point-wise application of the same copula to $\mathcal{M}_1$ and $\mathcal{M}_2$. After all, such a relation is known to hold for the product copula [22], as well as for all copulas in the case of p-boxes [9]. It is however shown in this same later

paper that this inclusion is generally not true. The next counter-example shows that this is also the case for possibility distributions. Consider the same distributions

$$\begin{array}{rclcrcl} m_X([\overline{x},\overline{x}]) & = & m_1^X\,, & & m_Y([\overline{y},\overline{y}]) & = & m_1^Y\,, \\ m_X([\overline{x}-\delta_x,\overline{x}+\delta_x]) & = & m_2^X\,, & & m_Y([\overline{y}-\delta_y,\overline{y}+\delta_y]) & = & m_2^Y\,. \end{array}$$

as before. Let us now consider the copula $C = \min$, giving when we apply it as in Equation 5 the joint mass

$$\begin{array}{rcl} m_{XY}^C([\overline{x},\overline{x}]\times[\overline{y},\overline{y}]) & = & \min(m_1^X, m_1^Y)\,, \\ m_{XY}^C([\overline{x}-\delta_x,\overline{x}+\delta_x]\times[\overline{y},\overline{y}]) & = & m_1^Y - \min(m_1^X, m_1^Y)\,, \\ m_{XY}^C([\overline{x},\overline{x}]\times[\overline{y}-\delta_y,\overline{y}+\delta_y]) & = & m_1^X - \min(m_1^X, m_1^Y)\,, \\ m_{XY}^C([\overline{x}-\delta_x,\overline{x}+\delta_x]\times[\overline{y}-\delta_y,\overline{y}+\delta_y]) & = & 1 - m_1^X - m_1^Y + \min(m_1^X, m_1^Y)\,. \end{array}$$

If we compute the belief of the event $\{\overline{x}\}\times\{\overline{y}\}$, this gives

$$\mathrm{Bel}(\{\overline{x}\}\times\{\overline{y}\}) \;=\; m_{XY}^C([\overline{x},\overline{x}]\times[\overline{y},\overline{y}]) = \min(m_1^X, m_1^Y)\,. \qquad (14)$$

Now, let us consider the two probability distributions $p_X(\overline{x}) = m_1^X, p_X(\overline{x}+\delta_x) = m_2^X$ and $p_Y(\overline{y}) = m_1^Y, p_Y(\overline{y}-\delta_y) = m_2^Y$. Applying copula $C = \min$ to this gives the joint probability:

$$\begin{array}{rcl} p_{XY}^C(\{\overline{x}\}\times\{\overline{y}-\delta_y\}) & = & \min(m_1^X, m_2^Y)\,, \\ p_{XY}^C(\{\overline{x}\}\times\{\overline{y}\}) & = & m_1^X - \min(m_1^X, m_2^Y)\,, \\ p_{XY}^C(\{\overline{x}+\delta_x\}\times\{\overline{y}-\delta_y\}) & = & m_2^Y - \min(m_1^X, m_2^Y)\,, \\ p_{XY}^C(\{\overline{x}+\delta_x\}\times\{\overline{y}\}) & = & 1 - m_1^X - m_2^Y + \min(m_1^X, m_2^Y)\,. \end{array}$$

The joint value $p_{XY}^C(\{\overline{x}\}\times\{\overline{y}\}) = m_1^X - \min(m_1^X, m_2^Y)$ can be lower than Equation (14), depending on the values of m_1^X, m_1^Y, m_2^Y. For instance taking $m_1^X = 0.1, m_1^Y = 0.9, m_2^Y = 0.1$ yields:

$$\begin{array}{rcl} p_{XY}^C(\{\overline{x}\}\times\{\overline{y}\}) = & 0.1 - \min(0.1, 0.1) & = 0\,, \\ \mathrm{Bel}(\{\overline{x}\}\times\{\overline{y}\}) = & \min(0.1, 0.9) & = 0.1\,. \end{array}$$

This shows that there will be in general no inclusion between the joint models obtained from a mass-wise and a point-wise applications of copulas in the case of possibility distributions. Another example of this can be found in Example 6 from [9], which also contains additional insights on the inclusion when the copula verifies symmetric properties.

5. Using Copulas to Propagate Uncertainty

5.1. Propagating the Uncertainty using Multivariate Belief Functions

Let $f : \mathcal{X} \otimes \mathcal{Y} \to \mathcal{Z}$ be a mapping and we define the random variables Z as $Z = f(X, Y)$. In a precise setting, the probability mass distribution p_Z of Z is:

$$\forall z \in \mathcal{Z}, p_Z(z) = \sum_{\substack{x,y \\ z=f(x,y)}} p(x, y)\,. \tag{15}$$

Determining every (x, y), whose image by f equals z, is not always trivial. This becomes even more complex when we are considering copulas with $N > 2$ variables. Note that the H-volume is a sum of 2^N terms, which also increases exponentially with the dimension. In the continuous case, the H-volume is replaced with the density h of the joint CDF, and the density of Z is:

$$p_Z(z) = \int_{\mathcal{X}} \int_{\mathcal{Y}} h(x, y)\mathrm{Ind}(f(x, y) = z)dxdy\,, \tag{16}$$

where Ind is an indicator function.

The same principle can be used with belief functions instead of precise probability distributions. Given the joint mass distribution function constructed with the copula, it is then possible to compute the mass distribution function of a random set Z:

$$\forall a^Z \subseteq \mathcal{Z}, m_Z(a^Z) = \sum_{\substack{a_i^X, a_j^Y \\ a^Z = f(a_i^X, a_j^Y)}} m_{XY}(a_i^X, a_j^Y)\,. \tag{17}$$

As in the precise case, computing the image of f for every pair of focal sets (a_i^X, a_j^Y) is not always trivial.

5.2. Uncertainty Propagation through the SAD Cost Function

To illustrate how to propagate the uncertainty using belief functions and a copula, we will present in this section the case of the uncertainty related to the SAD cost function.

The SAD is computed between two 3×3 windows W_L, W_R. We use the mass distribution m_p of Equation (7) to represent the uncertainty of each pixel p. For every pair of pixels $p \in I_L, q \in I_R$, we note $\mathrm{AD}_{pq} = |i_p - i_q|$. There exists 3 focal sets related to the AD:

- a_1^{AD} is obtained by computing the AD of a_1^p and a_1^q
- a_2^{AD} is obtained by computing the AD of a_2^p and a_1^q or a_1^p and a_2^q
- a_3^{AD} is obtained by computing the AD of a_2^p and a_2^q

To compute their exact image through the AD, we need to take into account the non-monotonicity of the absolute value around 0:

$$\begin{aligned}
a_1^{\mathrm{AD}} &= [\![\mathrm{AD}_{pq},\ \mathrm{AD}_{pq}]\!]\,,\\
a_2^{\mathrm{AD}} &= [\![\mathrm{AD}_{pq}-1,\ \mathrm{AD}_{pq}+1]\!] \text{ if } \mathrm{AD}_{pq} > 0\,,\\
&= [\![\mathrm{AD}_{pq},\ \mathrm{AD}_{pq}+1]\!] \text{ otherwise }\ ,\\
a_3^{\mathrm{AD}} &= [\![\mathrm{AD}_{pq}-2,\ \mathrm{AD}_{pq}+2]\!] \text{ if } \mathrm{AD}_{pq} > 1\,,\\
&= [\![\mathrm{AD}_{pq}-1,\ \mathrm{AD}_{pq}+2]\!] \text{ if } \mathrm{AD}_{pq} = 1\,,\\
&= [\![\mathrm{AD}_{pq},\ \mathrm{AD}_{pq}+2]\!] \text{ otherwise}\,.
\end{aligned}$$

Obtaining the final SAD is easier as the bounds of each interval can be simply summed. The mass m_{SAD} of each focal set of the SAD is computed using Equations (13) and (17). The dependency between all of the pixels is represented by a Gaussian 9-copula.

5.3. Reducing the Computation Time of Exact Propagation

Determining the SAD focal sets bounds is an easy task. Computing the joint mass over two 3×3 windows is not as easy. The H-volume is now computed for a 18-copula, which is a sum of 2^{18} terms. Because the uncertainty of each pixel is represented by 2 focal sets, there are 2^{18} combinations of focal sets to evaluate for the two 3×3 windows. A way of reducing the computation time is to take advantage of the fact that the focal sets derived from a possibility distribution (or equivalently from its necessity measure) form a nested family of sets. In the general case, propagating two necessity measures $\mathrm{Nec}_X : \mathcal{P}(\mathcal{X}) \to [0,1]$, $\mathrm{Nec}_Y : \mathcal{P}(\mathcal{Y}) \to [0,1]$ through a mapping $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ with a copula C does not yield a necessity measure but "only" a belief function. For the special case where $\mathrm{Nec}_X, \mathrm{Nec}_Y$ are defined by symmetric uni-modal possibility distributions (typically triangular possibilities), and f is a monotone function applied to a linear combination $\alpha X + \beta Y + \gamma$ of X and Y, $(\alpha, \beta, \gamma) \in \mathbb{R}^3$, then the focal sets of $Z = f(X, Y)$ form a nested family of sets, which is characteristic of necessity measures [23]. Indeed, the focal sets of $\mathrm{Nec}_X, \mathrm{Nec}_Y$ are families of nested sets that can

be written as $([\overline{X} \pm \Delta x_i])$, $([\overline{Y} \pm \Delta y_j])$, with $\overline{X} \in \mathcal{X}, \overline{Y} \in \mathcal{Y}$ and $(\Delta x_i), (\Delta y_j)$ positive scalars. The focal sets of the linear combination of $\alpha X + \beta Y + \gamma$ are of the form:

$$\begin{aligned} a_{ij} = [\alpha\overline{X} + \beta\overline{Y} + \gamma - (|\alpha|\Delta x_i + |\beta|\Delta y_j), \\ \alpha\overline{X} + \beta\overline{Y} + \gamma + (|\alpha|\Delta x_i + |\beta|\Delta y_j)] \,, \end{aligned}$$

and is a nested family of sets. Applying a monotone function to those focal sets will keep the nesting property (but not their symmetry), which results in Bel_Z being a necessity measure. For more advanced functions such as multiplication, exponentials, sigmoids, etc..., this property is not always true (it is easy to find examples where the nesting property is not retained for Bel_Z), as shows the next example.

Example 1. *Consider the function $f(x, y) = (x^2 + 1) + (y^2 + 1)$, and the following mass functions*

$$\begin{array}{rclcrcl} m_X([0]) & = & m_1^X \,, & & m_Y([1]) & = & m_1^Y \,, \\ m_X([-2,2]) & = & m_2^X \,, & & m_Y([0,2]) & = & m_2^Y \,. \end{array}$$

After propagating through the product function, one gets:

$$\begin{array}{rrl} m_{XY}^C(f([0],[1])) = & m_{XY}^C([3]) & = C(m_1^X, m_1^Y) \,, \\ m_{XY}^C(f([-2,2],[1])) = & m_{XY}^C([3,7]) & = m_1^Y - C(m_1^X, m_1^Y) \,, \\ m_{XY}^C(f([0],[0,2])) = & m_{XY}^C([2,6]) & = m_1^X - C(m_1^X, m_1^Y) \,, \\ m_{XY}^C(f([-2,2],[0,2])) = & m_{XY}^C([2,10]) & = 1 - m_1^X - m_1^Y + C(m_1^X, m_1^Y) \,. \end{array}$$

The output focal sets do not form a nested family, and thus the mass function m_{XY}^C defines a belief function that is not a necessity function.

This property can be used to simplify and reduce the computations of focal sets bounds and their masses in the case where all of the AD are superior to 2 (to avoid the non-monotonicity of the absolute value around 0). In the case where the focal sets are nested, it holds that for every focal sets a_i^X, a_j^Y of $\mathrm{Nec}_X, \mathrm{Nec}_Y$ [21]:

$$\mathrm{Bel}_{XY}(a_i^X, a_j^Y) = C(\mathrm{Nec}_X(a_i^X), \mathrm{Nec}_Y(a_j^Y)) \,, \tag{18}$$

hence the joint belief function can be computed solely with the marginal masses of Nec_X and Nec_Y and the copula. There is no need to compute the joint mass, thus avoiding the computation of the H-volume with (10).

5.4. Reducing the number of operations during simulation

When joining multiple sources of uncertainty, the number of operations needed to compute the output mass function m_{out} grows exponentially. For instance, let us consider n sources of uncertainty, each represented by k focal sets. Here are some factors that can generate difficulties when trying to compute m_{out}:

- there are k^n output focal sets for which the mass m_{out} needs to be computed.
- Every value of m_{out} is being computed using the H-volume of a n-copula. The H-volume is the sum of 2^n evaluation of its copula.
- For copulas for which only the density is known, such as the Gaussian copula, each evaluation of the copula is a result of the integration of a n variate function. This can quickly become quite costly.

In our experiments, the images have a size of 375×450 and the disparity interval is $[-60, 0]$. We therefore compute more that 10^7 belief functions. Each belief function is the result of joining the uncertainty of $n = 2 \times 3 \times 3 = 18$ pixels, possessing only $k = 2$ distinct focal sets. In total, more than $10^7 \times (2 \times 2)^{18}$ integration of 18 variate functions needs to be integrated, which takes large amounts of time, even when maximizing the parallelization of the code. To give an order of magnitude: computing a gaussian copula of dimension 18 takes around 20 seconds on a AMD EPYC 7713 64-Core Processor with 2GHz, using Python and the SciPy library. It is however possible to take advantage of some properties of our models and of the copula to drastically reduce the computation time.

Firstly, if we can split the variables in multiple independent sets, then the H-volume can be computed more efficiently. This is due to the following property:

Proposition 1. *If a n-copula C can be expressed as the product of a k-copula C' and a $n-k$-copula C'', then the H-volume of C is the product of the H-volume of C' and the H-volume of C''.*

Proof. Let $X_1, \ldots, X_n$ be n real random variables, and C a n copula representing their dependency. Let assume that there is a $k \in \mathbb{N}$ such that $X_1, \ldots, X_k$ are

independent from $X_{k+1}, \ldots, X_n$. Then there exists to copulas C', C'' such that:

$$C(X_1, \ldots, X_n) = C'(X_1, \ldots, X_k) \times C''(X_{k+1}, \ldots, X_n)\,.$$

Let $(\underline{x}_i, \overline{x}_i)_{1 \leqslant i \leqslant n} \in \mathbb{R}^{2n}$ such that $\forall i \in [\![1, n]\!]$, $\underline{x}_i \leqslant \overline{x}_i$. Computing the product of the cost volume of C' and C'' yields:

$$\begin{aligned}
H'^{\overline{x}_1, \ldots \overline{x}_k}_{\underline{x}_1, \ldots \underline{x}_k} \times H''^{\overline{x}_{k+1}, \ldots \overline{x}_n}_{\underline{x}_{k+1}, \ldots \underline{x}_n} &= \left(\sum_{(x_i) \in \prod_{i=1}^{k} \{\underline{x}_i, \overline{x}_i\}} (-1)^{\#\{x_i \mid x = \overline{x_i}\}} C'(x_1, \ldots, x_k) \right) \\
&\times \left(\sum_{(x_j) \in \prod_{j=k+1}^{n} \{\underline{x}_j, \overline{x}_j\}} (-1)^{\#\{x_j \mid x = \overline{x_j}\}} C''(x_{k+1}, \ldots, x_n) \right) \\
&= \sum_{(x_i) \in \prod_{i=1}^{k} \{\underline{x}_i, \overline{x}_i\}} \times \sum_{(x_j) \in \prod_{j=k+1}^{n} \{\underline{x}_j, \overline{x}_j\}} (-1)^{\#\{x_i \mid x = \overline{x_i},\ i \leqslant k\}} \\
&\quad \times (-1)^{\#\{x_j \mid x = \overline{x_j},\ j > k\}} C'(x_1, \ldots, x_k) C''(x_{k+1}, \ldots, x_n) \\
&= \sum_{(x_i) \in \prod_{i=1}^{n} \{\underline{x}_i, \overline{x}_i\}} (-1)^{\#\{x_i \mid x = \overline{x_i}\}} C'(x_1, \ldots, x_k) \\
&\quad \times C''(x_{k+1}, \ldots, x_n) \\
&= H^{\overline{x}_1, \ldots \overline{x}_n}_{\underline{x}_1, \ldots \underline{x}_n}
\end{aligned}$$

□

Extending this result to any number of independent subsets of $\{X_1, \ldots, X_n\}$ is straightforward. This allows to reduce the computation of the H-volume for high dimensions. Take as an example a split into two sets of j and $j - n$ elements. The H-volume is now computed by evaluating a j-copula 2^j times, and a $n - j$ copula 2^{n-j} times instead of a n copula 2^n times. For density based copulas such as the gaussian copula, it also shifts the problem from integrating a n dimension function into the integration of two lower dimension functions. Using an AMD EPYC 7713 64-Core Processor with 2GHz and the SciPy library for Python, it takes 20 seconds to compute a 18-gaussian-copula but only around 1 second to compute a 9-gaussian-copula.

Another way of reducing the computation time in our case is to notice

that the gaussian copula used with correlation matrices of type:

$$\begin{pmatrix} 1 & \dots & \sigma & \dots & \sigma \\ \vdots & & \vdots & & \vdots \\ \sigma & \dots & 1 & \dots & \sigma \\ \vdots & & \vdots & & \vdots \\ \sigma & \dots & \sigma & \dots & 1 \end{pmatrix},$$

are exchange/permutation symmetrical (i.e., the order between variables inside the copula does not matter). When every univariate focal sets have the same mass distribution functions, then:

$$C(m_1, m_1, m_2) = C(m_1, m_2, m_1) = C(m_2, m_1, m_1)\,.$$

For every pixel of the same cluster of j pixels, we thus only need to evaluate the copula $j + 1$ times instead of 2^j.

6. Resulting Envelopes and Monte Carlo Simulations

We have presented the tools used for propagating belief functions through the cost function. This section details the construction of the correlation matrix for the Gaussian copula, and how envelopes were computed from the propagated belief function. We also generate samples from different noise models with a dependency specified by the product and Gaussian copulas. This noise is added to the input images, and we apply the Monte Carlo method to generate multiple "noised" cost curves. Those curves are compared to the envelopes to determine if the propagated belief functions is able to correctly characterize the uncertainty.

6.1. Sampling from Copulas

In our case, we want to model with a copula the dependency between the random intensities of two pixels: one in the left image, and one in the right image. We propose to model their dependency with the product copula if the pixels are not from the same physical object (meaning that the value of their intensities are independent), and by a Gaussian copula with a covariance matrix $\left(\begin{smallmatrix} 1 & \sigma_{\text{obj}} \\ \sigma_{\text{obj}} & 1 \end{smallmatrix}\right)$, $\sigma_{\text{obj}} \in \mathbb{R}^+$, if they belong to the same physical object in the scene. A segmentation of the image based on the ground truth of the disparity is used to determine if two pixels belong to the same object.

As we validate our method using Monte Carlo simulations in Section 6, we need to be able to sample random vectors from a copula. We now detail a method for sampling from copulas in general and a method for sampling from the Gaussian copula. Given a copula C, and two cumulative distribution functions F_X and F_Y, the method used to generate a pair of observations (x, y) from a joint CDF $C(F_X, F_Y)$ is the following:

- Sample two independent samples u_1, u_2 from a uniform distribution on [0,1]
- Set $v = \partial C^{-1}(u_2)$ where ∂C^{-1} is the quasi-inverse of the partial derivative of C regarding its first variable (which exists almost everywhere and is invertible).
- (u_1, v) each follow a uniform distribution on $[0, 1]$, and their associated copula is C
- The desired pair is $(x_1, x_2) = (F_X^{-1}(u_1), F_Y^{-1}(v))$, with F_X^{-1}, F_Y^{-1} being the quasi-inverses of the marginals CDFs.

Details of this method in the N-dimensional case can be found in [24]. Simulation draws from the Gaussian N-copula with correlation matrix R are simpler to obtain:

- Compute the Cholesky decomposition A of the correlation matrix R
- Draw N independent random samples $u = (u_1, \ldots, u_N)'$ from $\mathcal{N}(0, 1)$
- Set $v = Au$
- Set $w_k = \Phi(v_k)$ where Φ is the univariate normal cumulated distribution function
- The desired draw is $(x_1, \ldots, x_N) = (F_1^{-1}(w_1), \ldots, F_N^{-1}(w_N))$ with F_k^{-1}, being the quasi-inverse of the k-th marginal CDF.

6.2. Constructing a Correlation Matrix for the Gaussian Copula

Section 6.1 presented our aim to sample from copulas to validate the computed envelopes. The current section details the construction of the Gaussian copula which will be used. The Gaussian copula is parameterized by a correlation matrix R. The correlation between the uncertain sources

is based on a segmentation $S : (I_L \cup I_R) \to [\![0, K]\!]$, $K \in \mathbb{N}$, of the images, performed on the ground truth disparity of each pixel. Given two pixels $(p, q) \in (I_L \cup I_R)^2$, their covariance is determined by:

$$\mathrm{cov}(p, q) = \begin{cases} 1 & \text{if } p = q\,, \\ \rho_k, & \text{if } p \neq q \text{ and } S(p) = S(q)\,, \\ 0, & \text{otherwise}\,. \end{cases} \tag{19}$$

The segmentation contains $K = 8$ different clusters, and for every k in $[\![0, K]\!]$, ρ_k is assigned an arbitrary value between 0.9 and 1.

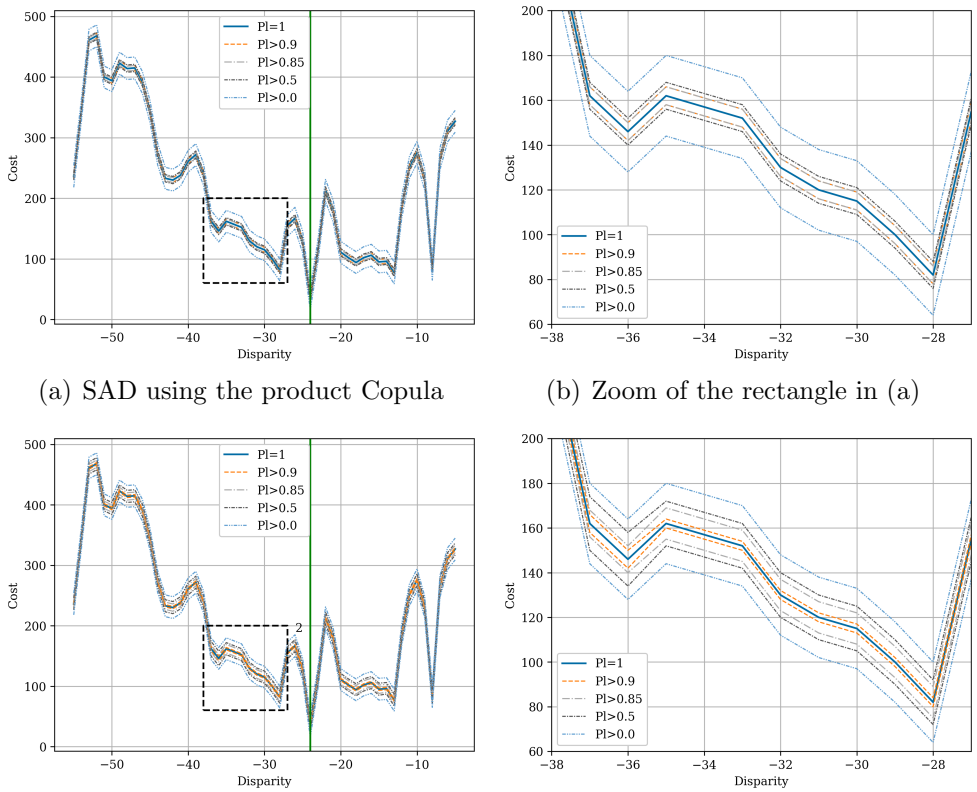
We can thus build a correlation matrix R for all pixels of $(I_L \cup I_R)$, and sample a perturbation on all of the pixels using a Gaussian copula with this correlation matrix. The Gaussian $2-$copula and Gaussian $9-$copula are used to model the dependency between masses in the uncertainty propagation step and their correlation matrix is also constructed using Equation (19).

6.3. Envelopes from the Propagated Belief Function

To illustrate the propagation of uncertainty with a concrete example, we used the *Middlebury* dataset[1], consisting of pairs of left-right images with the correct disparity map available. An example of an image pair from this dataset is presented in Figure 1. For every pixel in the left image, we computed its SAD cost curve, while propagating the uncertainty model presented in (3.2). The focal sets representing the uncertainty related to the SAD value at every considered disparity are intervals containing the "precise" SAD value. Upper and lower envelopes with different plausibility levels γ have been computed. They represent the biggest (resp. lowest) focal set bound $\overline{a}$ whose plausibility, computed using Equation (4), is above a given threshold γ: $\mathrm{Pl}(\overline{a}) \geqslant \gamma$. There are thus two bounds for every plausibility level γ. The plausibility threshold 0 is strict $\mathrm{Pl}(\overline{a}) > 0$ and represents the support of the SAD. The plausibility threshold 1 coincides with the value of the cost curve obtained in the "precise setting" (i.e. without considering the uncertainty).

Figure 4 contains different plausibility levels of the cost curve computed with the product copula C_Π and a gaussian copula C_R. The support of the SAD ($\mathrm{Pl} > 0$) is the same for both copulas, as both copulas assign non-null masses to the same focal sets. The plausibility levels are different when

[1] `https://vision.middlebury.edu/stereo/data/scenes2003/`



(a) SAD using the product Copula

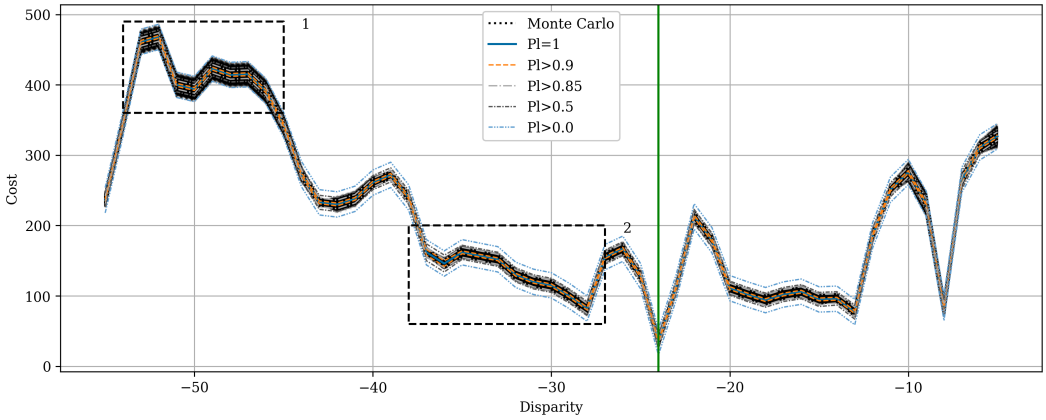
(b) Zoom of the rectangle in (a)

(c) SAD using the Gaussian Copula

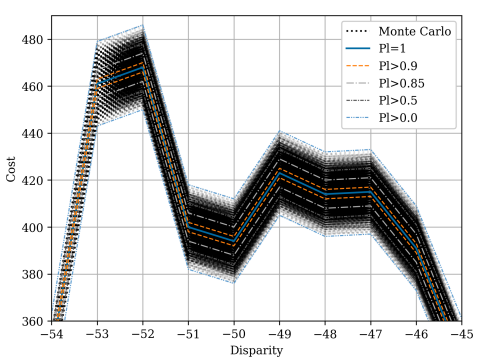
(d) Zoom of the rectangle in (c)

Figure 4: Plausibility levels of a cost curve for the product copula C_Π and the gaussian copula C_R. The green vertical line represents the true disparity. The figure on the right are a zoom of the black dashed rectangle.

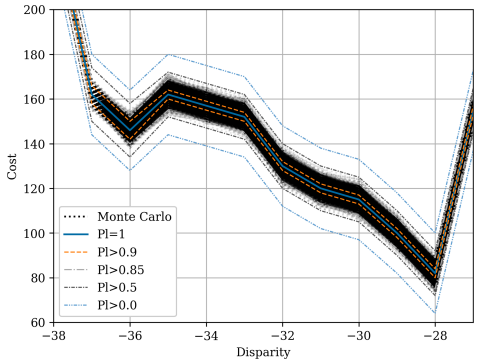
considering different copulas, except for $\gamma = 1$. This can be observed on figures 4(b) and 4(d). In the specific case of the product copula C_Π, the joint mass is easy to compute as it is simply the product of the marginal masses. As stated previously, the envelopes corresponding to plausibility levels 0 and 1 are the same for both copulas. However, the plausibility levels 0.85 and 0.5 are more concentrated around plausibility level 1 in the case of the product copula than in the case of the Gaussian copula. The inverse observation can be done for plausibility level 0.9. By construction, the Gaussian copula C_R is more comonotone than the product copula, which could explain this effect.



(a) Plausibility levels and Monte Carlo sampling using a gaussian copula

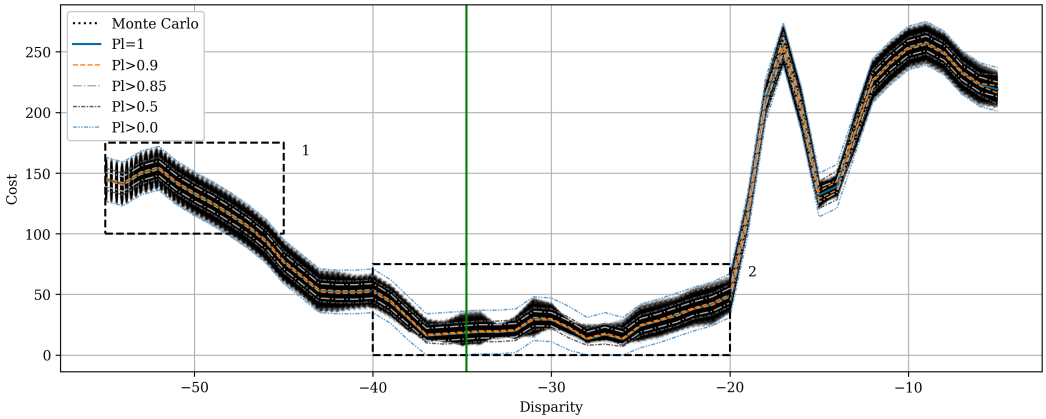


(b) Zoom over the first rectangle

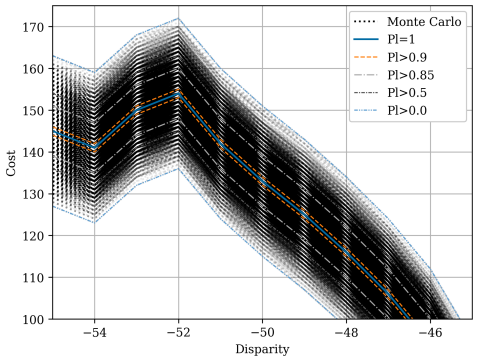


(c) Zoom over the second rectangle

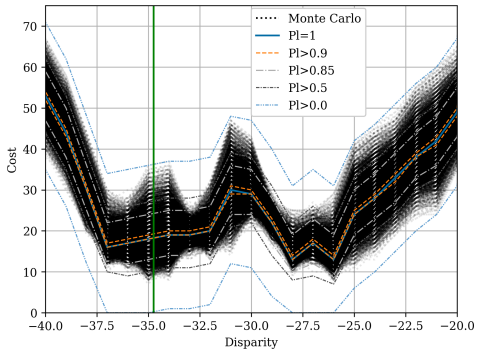
Figure 5: Plausibility levels and Monte Carlo sampling for a pixel of the left image



(a) Plausibility levels and Monte Carlo sampling using a gaussian copula



(b) Zoom over the first rectangle



(c) Zoom over the second rectangle

Figure 6: Plausibility levels and Monte Carlo sampling for a pixel of the left image

6.4. Comparison with Monte Carlo Samplings

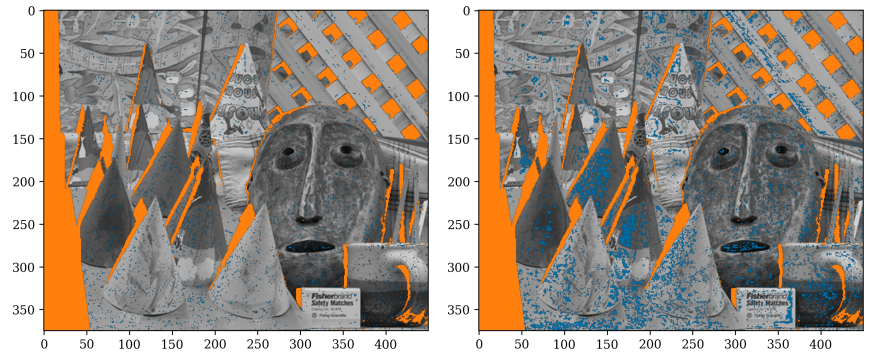
Figure 5 and 6 contains plausibility levels and Monte Carlo draws sampled from the Gaussian copula as in Section 6.1, with marginals uniformly sampled from the credal sets of 3.2. Different plausibility levels computed with their respective copula are also plotted. We observe that all Monte Carlo samplings are correctly contained in the support envelopes. This holds for both the product and the Gaussian copulas. It appears in Figure 6(c) that although the different plausibility levels seem to correctly contain the Monte Carlo samples, they sometimes fail to correctly grasp the fluctuations of the dispersion of the samples. For instance the Monte Carlo draws are first dense around disparity -37, then seem to spread around -35, and finally regather around disparity -32. The plausibility envelopes are more regular around those disparities. This illustrates that the joint credal set computed is different from the "true" point-wise credal set described in section 4.3.2. Despite the differences between those sets, the Monte Carlo sampling suggests that joining belief functions using a copula can correctly estimate the point-wise credal set, as in Figure 5(c) or 6(c). A quantitative analysis of this estimation is presented in Table 1, where the coverage, i.e. the proportion of Monte Carlo samplings that are contained inside the plausibility envelopes, is computed for different plausibility levels γ. First and second rows of Table 1 represent the average coverage for pixels in Figures 5 and 6 respectively, while the last row represent the average global over the whole left image. The fact that the coverage is always 100% for $\gamma = 0$ indicates that every sample is contained inside the support envelopes. Because we defined the envelopes as lower and upper bounds of the focal sets with a plausibility superior to γ, the coverage should be superior to $1 - \gamma$, which is indeed the case.

$p_L = (i, j)$	$\gamma = 0.9$	$\gamma = 0.85$	$\gamma = 0.5$	$\gamma = 0$
(100, 120)	$64,5\%$	$94,5\%$	$99,0\%$	100%
(200, 150)	$30,0\%$	$82,6\%$	$95,2\%$	100%
Global	$41,1\%$	$87,6\%$	$96,8\%$	100%

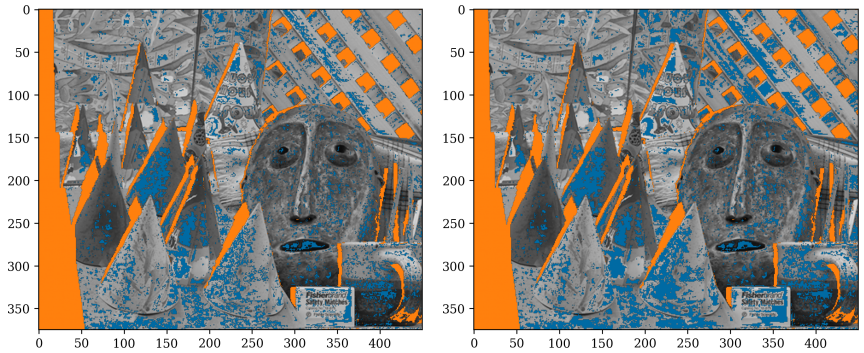
Table 1: Average coverage for different pixels p_L of the left image and plausibility levels γ

6.5. Potential improvement for stereo-algorithms

Computing confidence envelopes for the cost volume provides additional information on potential matches. This section presents observations supporting



(a) $\gamma = 0.9,\ \Delta s_\gamma = 4.05\%$ (b) $\gamma = 0.85,\ \Delta s_\gamma = 16.11\%$



(c) $\gamma = 0.5,\ \Delta s_\gamma = 21.41\%$ (d) $\gamma = 0,\ \Delta s_\gamma = 28.87\%$

Figure 7: Spatial disposition of potential improvements for different values of γ. Pixels where improvements can occur appear in blue. Pixels in occluded regions appear in orange. The background contains the grayscale left image.

the idea that this information has the potential to improve the results of stereo-matching algorithms.

A popular method for evaluating the performances of stereo-algorithms is to compute the proportion of pixels (i, j) for which the distance between the true disparity and the predicted disparity is less than a pixel. By denoting $d_{\text{true}}(i, j)$ the true disparity and $\tilde{d}(i, j) = \text{argmin}_d \text{SAD}(i, j, d)$ the estimated disparity at position (i, j), this score s is expressed as:

$$s = \frac{\#\{(i, j)\ s.t.\ |d_{\text{true}}(i, j) - \tilde{d}(i, j)| < 1\}}{\#\{(i, j)\}}\,. \tag{20}$$

Furthermore, for a given plausibility value $\gamma \in [0, 1]$, and for a pixel (i, j) we

can define sets of potential disparities $D_\gamma^{i,j}$ as:

$$D_\gamma^{i,j} = \{d \mid \underline{\mathrm{SAD}}_\gamma(i,j,d) \leqslant \overline{\mathrm{SAD}}_\gamma(i,j,\tilde{d})\}\,. \tag{21}$$

This is the set of all disparities for which the lower estimation of SAD_γ for plausibility γ is inferior to the upper estimation of SAD_γ for the predicted disparity. To see if this set contains relevant information, we wan compute the optimal s_1 score that could be obtain using this set of possible disparities:

$$s_\gamma^{opt} = \frac{\#\{(i,j)\ s.t.\ \min_{d\in D_\gamma^{i,j}} |d_{\mathrm{true}}(i,j) - d(i,j)| < 1\}}{\#\{(i,j)\}}\,. \tag{22}$$

We define the potential gain as $\Delta s_\gamma = s_\gamma^{opt} - s$. Example of optimal scores and potential gain can be found in Table 2. We can see that the potential gain for $\gamma = 0.9$ is small, but increases rapidly for lower values of γ. We consider that a pixel at position (i,j) benefits from the method if:

$$|d_{\mathrm{true}}(i,j) - \tilde{d}(i,j)| \geqslant 1 \quad \text{and} \quad \min_{d\in D_\gamma^{i,j}} |d_{\mathrm{true}}(i,j) - d(i,j)| < 1\,. \tag{23}$$

In plain words, a pixel can benefit from the method if the estimated disparity is more than one pixel away from the true disparity, but there is a pixel in $D_\gamma^{i,j}$ that is less than a pixel away from the true disparity. Figure 6.5 displays the spatial disposition of pixels that can benefit from our method. Pixels in occluded regions who do not have a match appear in orange. Pixels where improvement can occur appear in blue. We can see that the pixels with potential improvements are logically positioned in homogeneous areas, where many disparities have low matching costs (as in Figure 6(a)).

$s = 52.87\%$	$\gamma = 0.9$	$\gamma = 0.85$	$\gamma = 0.5$	$\gamma = 0$
s_γ^{opt}	56.92%	66.99%	74.28%	81.75%
Δs_γ	4.05%	16.11%	21.41%	28.87%

Table 2: Optimal score and potential gain for different plausibility γ

7. Conclusion

This contribution presents a real-life application of uncertainty propagation using possibility distributions as models, and copulas to characterize the

dependency between different random sources. In order to propagate the uncertainty in the matching step of a photogrammetry 3D pipeline, we introduce the use of cost functions, and presented a simple model to represent the sources of uncertainty in input images. The different steps for propagating the uncertainty are detailed for didactic purposes, with the intention of highlighting the potential of using imprecise models in concrete cases. Additionally, a sufficient condition for preserving possibilities (i.e., keeping the focal elements nested) after the propagation is proposed, which can be used to reduce the computational time.

Envelopes are deduced from the propagated plausibility functions, which correctly frame different types of input noise, generated using Monte Carlo simulations in our empirical studies. Although copulas do not bear the same meaning when used with precise density functions and when used on mass functions induced by belief functions, simulations show that the propagated belief functions allow us to generate correct envelopes which estimate the possible value of the cost function. We also show that dealing with uncertain cost functions has the potential to improve the results of stereo matching algorithms, albeit at the expense of additional uncertainty.

We presented a case of using imprecise probability to estimate the cost function in a photogrammetry 3D pipeline problem, yet did not consider the uncertainty stemming from the formulation of the function itself. In short, we have accounted for uncertainty in the input, and not in the model. It is however not guaranteed that comparing two patches of images as we did is sufficient to determine all disparities with certainty. A future work would consist in taking into account both the uncertainty on the input image and the uncertainty related to the cost function's ability to correctly distinguish homologous pixels. Indeed, this is influenced by many parameters, such as the choice of the cost function, the size and shape of the windows, the use of geometric regularization *etc.* Imprecise models could provide the necessary tools to correctly process this uncertainty. Another perspective is to extend this work to more complex cost functions and to other steps of the 3D pipeline, such as rasterisation of imprecise 3D point clouds or stereo-rectification of pairs of images.

References

[1] D. Scharstein, R. Szeliski, R. Zabih, A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, in: Proceedings IEEE

Workshop on Stereo and Multi-Baseline Vision (SMBV 2001), IEEE Comput. Soc, 2001, pp. 131–140. URL: http://ieeexplore.ieee.org/document/988771/. doi:10.1109/SMBV.2001.988771.

[2] X. Hu, P. Mordohai, A quantitative evaluation of confidence measures for stereo vision, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (2012) 2121–2133. URL: http://ieeexplore.ieee.org/document/6143951/. doi:10.1109/TPAMI.2012.46.

[3] M. Mehltretter, C. Heipke, Aleatoric uncertainty estimation for dense stereo matching via CNN-based cost volume analysis, ISPRS Journal of Photogrammetry and Remote Sensing 171 (2021) 63–75. URL: https://linkinghub.elsevier.com/retrieve/pii/S0924271620303026. doi:10.1016/j.isprsjprs.2020.11.003.

[4] E. Sarrazin, M. Cournet, L. Dumas, V. Defonte, Q. Fardet, Y. Steux, N. Jimenez Diaz, E. Dubois, D. Youssefi, F. Buffe, Ambiguity concept in stereo matching pipeline, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLIII-B2-2021 (2021) 383–390. URL: https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLIII-B2-2021/383/2021/. doi:10.5194/isprs-archives-XLIII-B2-2021-383-2021.

[5] M. Mehltretter, Uncertainty estimation for end-to-end learned dense stereo matching via probabilistic deep learning, 2020. URL: http://arxiv.org/abs/2002.03663. arXiv:2002.03663 [cs], number: arXiv:2002.03663.

[6] M. Okutomi, T. Kanade, A stereo matching algorithm with an adaptive window: theory and experiment, IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (1994) 920–932. URL: http://ieeexplore.ieee.org/document/310690/. doi:10.1109/34.310690.

[7] D. Dubois, H. Prade, When upper probabilities are possibility measures, Fuzzy Sets and Systems 49 (1992) 65–74. URL: https://linkinghub.elsevier.com/retrieve/pii/016501149290110P. doi:10.1016/0165-0114(92)90110-P.

[8] B. Schmelzer, Sklar's theorem for minitive belief functions, International Journal of Approximate Reasoning 63 (2015) 48–61. URL: https://linkinghub.elsevier.com/retrieve/pii/S0888613X15000821. doi:10.1016/j.ijar.2015.05.010.

[9] B. Schmelzer, Random sets, copulas and related sets of probability measures, International Journal of Approximate Reasoning (2023) 108952.

[10] R. Malinowski, S. Destercke, E. Dubois, L. Dumas, E. Sarrazin, Uncertainty propagation using copulas in a 3d stereo matching pipeline, in: International Symposium on Imprecise Probability: Theories and Applications, PMLR, 2023, pp. 288–298.

[11] A. Fusiello, E. Trucco, A. Verri, A compact algorithm for rectification of stereo pairs, Machine Vision and Applications 12 (2000) 16–22. URL: http://link.springer.com/10.1007/s001380050120. doi:10.1007/s001380050120.

[12] P. Fua, Combining stereo and monocular information to compute dense depth maps that preserve depth discontinuities, in: Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (II), 1991, pp. 1292–1298.

[13] A. Buades, G. Facciolo, Reliable multiscale and multiwindow stereo matching, SIAM Journal on Imaging Sciences 8 (2015) 888–915. URL: http://epubs.siam.org/doi/10.1137/140984269. doi:10.1137/140984269.

[14] M. J. Hannah, Computer Matching of Areas in Stero Images, phdthesis, Stanford University Computer Science Department, 1994. URL: https://apps.dtic.mil/sti/pdfs/AD0786720.pdf.

[15] R. Zabih, J. Woodfill, Non-parametric local transforms for computing visual correspondence, in: J.-O. Eklundh (Ed.), Computer Vision — ECCV '94, volume 801, Springer Berlin Heidelberg, 1994, pp. 151–158. URL: http://link.springer.com/10.1007/BFb0028345. doi:10.1007/BFb0028345, series Title: Lecture Notes in Computer Science.

[16] J. Žbontar, Y. LeCun, Stereo matching by training a convolutional neural network to compare image patches, Journal of Machine

Learning Research 17 (2016). URL: http://arxiv.org/abs/1510.05970. arXiv:1510.05970.

[17] J. Michel, E. Sarrazin, D. Youssefi, M. Cournet, F. Buffe, J.-M. Delvit, A. Emilien, J. Bosman, O. Melet, C. L'Helguen, A new satellite imagery stereo pipeline designed for scalability, robustness and performance, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences V-2-2020 (2020) 171–178. URL: https://www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net/V-2-2020/171/2020/. doi:10.5194/isprs-annals-V-2-2020-171-2020.

[18] M. Sklar, Fonctions de Répartition À N Dimensions Et Leurs Marges, Université Paris 8, 1959. URL: https://books.google.fr/books?id=nreSmAEACAAJ.

[19] R. B. Nelsen, An introduction to copulas, Springer series in statistics, 2. ed ed., Springer, 2006.

[20] S. Ferson, W. Oberkampf, W. Tucker, J. Zhang, L. Ginzburg, D. Berleant, J. Hajagos, R. Nelsen, Dependence in probabilistic modeling, Dempster-Shafer theory, and probability bounds analysis., Technical Report SAND2004-3072, 919189, Sandia National Laboratories, 2004. URL: https://www.osti.gov/servlets/purl/919189/. doi:10.2172/919189.

[21] R. Malinowski, S. Destercke, Copulas, lower probabilities and random sets: How and when to apply them?, in: Building Bridges between Soft and Statistical Methodologies for Data Science, volume 1433, Springer International Publishing, 2023, pp. 271–278. URL: https://link.springer.com/10.1007/978-3-031-15509-3_36. doi:10.1007/978-3-031-15509-3_36, series Title: Advances in Intelligent Systems and Computing.

[22] I. Couso, S. Moral, P. Walley, A survey of concepts of independence for imprecise probabilities, Risk, Decision and Policy 5 (2000) 165–181.

[23] G. Shafer, A Mathematical Theory of Evidence, 1976. URL: https://www.semanticscholar.org/

paper/A-mathematical-theory-of-evidence-Shafer/4cd91c51098783ec972f6a0ab430cacdd634a5b2.

[24] U. Cherubini, E. Luciano, W. Vecchiato, Copula Methods in Finance, 1 ed., Wiley, 2004. URL: https://onlinelibrary.wiley.com/doi/book/10.1002/9781118673331. doi:10.1002/9781118673331.